u-blox ZED-F9P

Interface Description

Including Receiver Description

Abstract

This is an early version of the u-blox ZED-F9P high precision positioning receiver Interface Description and Receiver Description

www.u-blox.com

UBX-18010853 - R04

Document Information		
Title	**u-blox ZED-F9P Interface Description**	
Subtitle	Including Receiver Description v27.00	
Document type	Manual	
Document number	UBX-18010853	
Revision and date	R04 (c355483)	6 July 2018
Document status	Objective Specification	

Document status explanation	
Objective Specification	Document contains target values. Revised and supplementary data will be published later.
Advance Information	Document contains data based on early testing. Revised and supplementary data will be published later.
Early Production Information	Document contains data from product verification. Revised and supplementary data may be published later.
Production Information	Document contains the final product specification.

Table of Contents

Preface

1 Document Overview

The Interface Description Including Receiver Description is an important resource for integrating and configuring u-blox receivers. This document has a modular structure and it is not necessary to read it from the beginning to the end. There are two main sections: The Receiver Description and the Interface Description.

The *Receiver Description* describes the software aspects of system features and configuration of u-blox receivers. The Receiver Description is structured according to areas of functionality, with links provided to the corresponding NMEA and UBX messages, which are described in the Interface Description.

The *Interface Description* is a reference describing the messages used by the u-blox receiver and is organized by the specific NMEA, UBX, and RTCM messages.

> ! *This document provides general information on u-blox receivers. Some information might not apply to certain products. Refer to the product Data Sheet and/or Hardware Integration Manual for possible restrictions or limitations.*

2 Firmware and Protocol Versions

The protocol version defines a set of messages that are applicable across various u-blox products. Each firmware used by a u-blox receiver supports a specific protocol version, which is not configurable.

The following sections will explain how to decode the shown information to get the firmware and the protocol version.

2.1 How to Determine the Version and the Location of the Firmware

The u-blox receiver can run a firmware from two different locations:

- Internal ROM
- External Flash memory

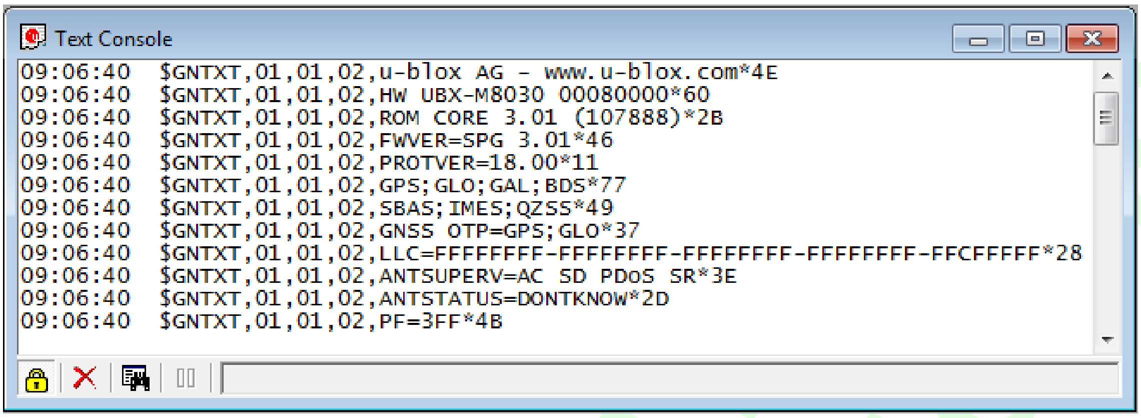
The location and the version of the currently running firmware can be found in the boot screen or in the UBX-MON-VER message.

For firmware supporting Protocol Version 24 and above:

- Boot screen, Protocol Version 24 and above
- `UBX-MON-VER, Protocol Version 24 and above`

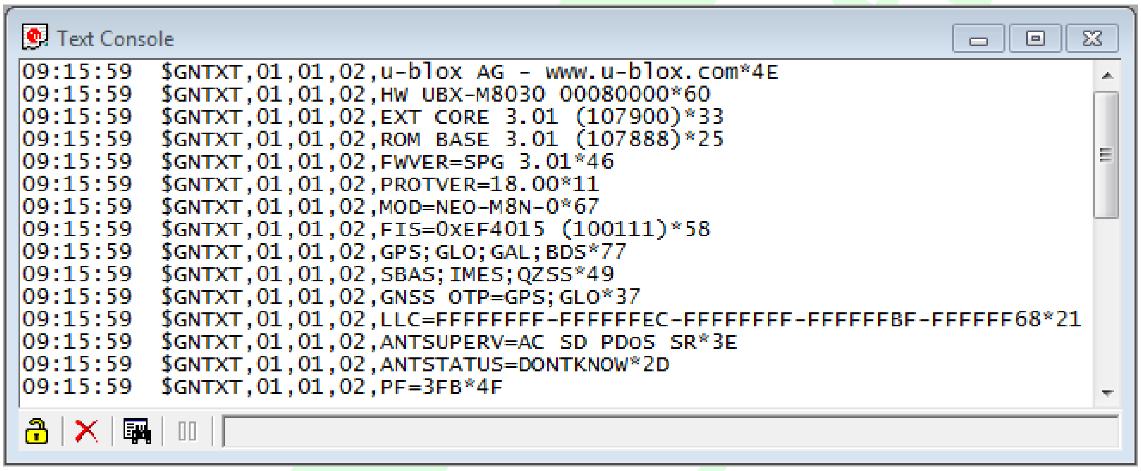
2.1.1 Decoding the Boot Screen (for Protocol Version 18 and Above)

Boot screen for a u-blox receiver running from ROM:



```
09:06:40  $GNTXT,01,01,02,u-blox AG - www.u-blox.com*4E
09:06:40  $GNTXT,01,01,02,HW UBX-M8030 00080000*60
09:06:40  $GNTXT,01,01,02,ROM CORE 3.01 (107888)*2B
09:06:40  $GNTXT,01,01,02,FWVER=SPG 3.01*46
09:06:40  $GNTXT,01,01,02,PROTVER=18.00*11
09:06:40  $GNTXT,01,01,02,GPS;GLO;GAL;BDS*77
09:06:40  $GNTXT,01,01,02,SBAS;IMES;QZSS*49
09:06:40  $GNTXT,01,01,02,GNSS OTP=GPS;GLO*37
09:06:40  $GNTXT,01,01,02,LLC=FFFFFFFF-FFFFFFFF-FFFFFFFF-FFFFFFFF-FFCFFFFF*28
09:06:40  $GNTXT,01,01,02,ANTSUPERV=AC SD PDoS SR*3E
09:06:40  $GNTXT,01,01,02,ANTSTATUS=DONTKNOW*2D
09:06:40  $GNTXT,01,01,02,PF=3FF*4B
```

Boot screen for a u-blox receiver running from Flash:



```
09:15:59  $GNTXT,01,01,02,u-blox AG - www.u-blox.com*4E
09:15:59  $GNTXT,01,01,02,HW UBX-M8030 00080000*60
09:15:59  $GNTXT,01,01,02,EXT CORE 3.01 (107900)*33
09:15:59  $GNTXT,01,01,02,ROM BASE 3.01 (107888)*25
09:15:59  $GNTXT,01,01,02,FWVER=SPG 3.01*46
09:15:59  $GNTXT,01,01,02,PROTVER=18.00*11
09:15:59  $GNTXT,01,01,02,MOD=NEO-M8N-0*67
09:15:59  $GNTXT,01,01,02,FIS=0xEF4015 (100111)*58
09:15:59  $GNTXT,01,01,02,GPS;GLO;GAL;BDS*77
09:15:59  $GNTXT,01,01,02,SBAS;IMES;QZSS*49
09:15:59  $GNTXT,01,01,02,GNSS OTP=GPS;GLO*37
09:15:59  $GNTXT,01,01,02,LLC=FFFFFFFF-FFFFFFEC-FFFFFFFF-FFFFFFBF-FFFFFF68*21
09:15:59  $GNTXT,01,01,02,ANTSUPERV=AC SD PDoS SR*3E
09:15:59  $GNTXT,01,01,02,ANTSTATUS=DONTKNOW*2D
09:15:59  $GNTXT,01,01,02,PF=3FB*4F
```

Not every line is output by every u-blox receiver in the boot screen. This depends on the product, the firmware location and the firmware version.

Possible lines in the boot screen and their meanings:

Entry	*Description*
`u-blox AG - www.u-blox.com`	Start of the boot screen
`HW UBX-M8030 00800000`	Hardware version of the u-blox receiver (u-blox M8 receiver)
`HW UBX-G8020 00800000`	Hardware version of the u-blox receiver (u-blox 8 receiver)
`ROM CORE 3.01 (107888)`	Firmware version 3.01 running from **ROM** (revision number)
`EXT CORE 3.01 (107900)`	Firmware version 3.01 running from **Flash** (revision number)
`ROM BASE 3.01 (107888)`	Underlying firmware version 3.01 in **ROM** (revision number)
`FWVER=SPG 3.01`	Firmware of product category and version where `SPG`: Firmware of Standard Precision GNSS product `HPG`: Firmware of High Precision GNSS product `ADR`: Firmware of ADR product `UDR`: Firmware of UDR product `TIM`: Firmware of Time Sync product `FTS`: Firmware of Time & Frequency Sync product
`PROTVER=18.00`	Supported protocol version
`MOD=NEO-M8N-0`	Module identification. Set in production.

Possible lines in the boot screen and their meanings: continued

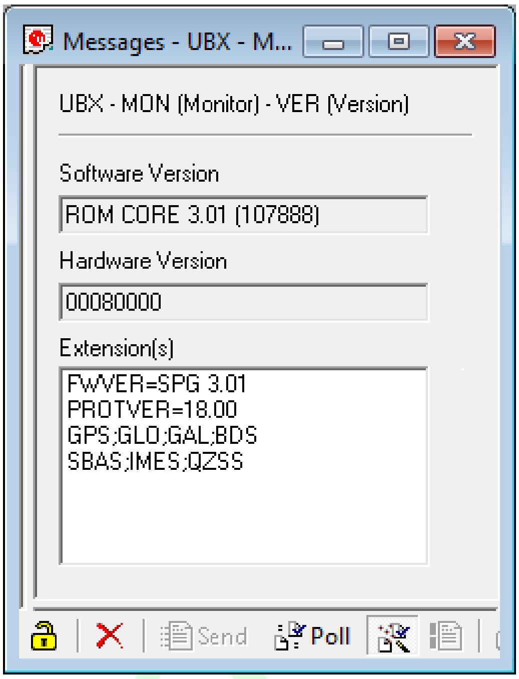
Entry	*Description*
`FIS=0xEF4015 (100111)`	Flash Information Structure (FIS) file for Flash memory with JEDEC `0xEF4015` found in the external flash memory. Revision number of the file is indicated in brackets.
`GPS;GLO;GAL;BDS`	Supported Major GNSS.
`SBAS;IMES;QZSS`	Supported Augmentation systems.
`GNSS OTP=GPS;GLO`	Default Major GNSS selection.
`LLC FFFFFFFF-FFFFFFFF-FFFFFFFF-FFFFFFFF-FFCFFFFF`	Low-level configuration of the u-blox receiver.
`ANTSUPERV=AC SD PDoS SR`	Configuration of the Antenna supervisor where `AC`: Active Antenna Control enabled `SD`: Short Circuit Detection enabled `OD`: Open Circuit Detection enabled `PDoS`: Short Circuit Power Down Logic enabled `SR`: Automatic Recovery from Short state
`PF=3FF`	Product configuration.

The line containing the `FWVER` *indicates which version of the firmware is currently running and is called* **firmware version** *in the rest of the document.*

The numbers in parentheses (revision numbers) should only be used to identify a known firmware version and are not guaranteed to increase over time.

2.1.2 Decoding the output of UBX-MON-VER (for Protocol Version 18 and above)

UBX-MON-VER for receiver running from ROM



UBX - MON (Monitor) - VER (Version)

Software Version

ROM CORE 3.01 (107888)

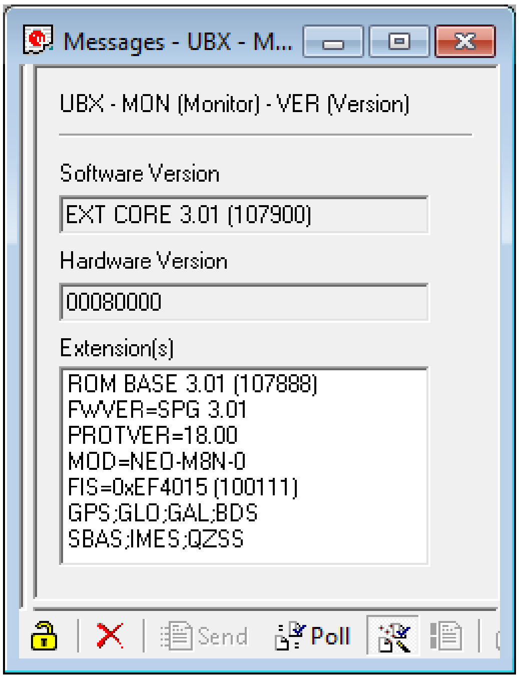
Hardware Version

00080000

Extension(s)

```
FWVER=SPG 3.01
PROTVER=18.00
GPS;GLO;GAL;BDS
SBAS;IMES;QZSS
```

UBX-MON-VER for receiver running from Flash



UBX - MON (Monitor) - VER (Version)

Software Version

EXT CORE 3.01 (107900)

Hardware Version

00080000

Extension(s)

```
ROM BASE 3.01 (107888)
FWVER=SPG 3.01
PROTVER=18.00
MOD=NEO-M8N-0
FIS=0xEF4015 (100111)
GPS;GLO;GAL;BDS
SBAS;IMES;QZSS
```

Possible fields in UBX-MON-VER and their meanings:

Entry	*Description*

Possible fields in UBX-MON-VER and their meanings: continued

Entry	*Description*

Software Version
`ROM CORE 3.01 (107888)`
`EXT CORE 3.01 (107900)`
	Currently running firmware version.
	If `ROM CORE`, then the u-blox receiver runs from **ROM**.
	If `EXT CORE`, then the u-blox receiver runs from **Flash**.

Hardware Version	The hardware version of the u-blox receiver.
Extension(s)	Extended information about the u-blox receiver firmware. See table below for the entries.

Not every entry is output by every u-blox receiver in the UBX-MON-VER extensions. This depends on the product, the firmware location and the firmware version.

Possible entries in UBX-MON-VER Extension(s):

Entry	*Description*
`ROM BASE 3.01 (107888)`	Underlying firmware version in ROM. If such an entry is present, then the u-blox receiver runs from **Flash**.
`FWVER=SPG 3.01`	Firmware of product category and version where `SPG`: Firmware of Standard Precision GNSS product `HPG`: Firmware of High Precision GNSS product `ADR`: Firmware of ADR product `UDR`: Firmware of UDR product `TIM`: Firmware of Time Sync product `FTS`: Firmware of Time & Frequency Sync product
`PROTVER=18.00`	Supported protocol version.
`MOD=NEO-M8N-0`	Module identification. Set in production.
`FIS=0xEF4015 (100111)`	Flash Information Structure (FIS) file for Flash memory with JEDEC `0xEF4015` found in the external flash memory. Revision number of the file is indicated in brackets.
`GPS;GLO;GAL;BDS`	Supported Major GNSS.
`SBAS;IMES;QZSS`	Supported Augmentation systems.

2.2 How to Determine the Supported Protocol Version of the u-blox Receiver

Each u-blox receiver reports its supported protocol version in the following ways:

- On start-up in the boot screen
- In the `UBX-MON-VER message`

with the line containing `PROTVER` (example: `PROTVER=18.00`).

Additionally, the *firmware string*, together with the *firmware version*, can be used to look up the corresponding protocol version. The tables below give an overview of the released firmware and their corresponding protocol versions.

2.2.1 u-blox 9 Firmware and Supported Protocol Versions

Firmware for High Precision GNSS Products

Firmware version	*Firmware string*	*Protocol Version*
HPG 1.00B03	EXT CORE 1.00 (554da8)	27.00

Receiver Description

3 Receiver Configuration

u-blox positioning receivers are fully configurable with UBX protocol messages. The configuration used by the receiver during normal operation is called the "current configuration". The current configuration can be changed during normal operation by sending UBX-CFG-VALSET messages over any I/O port. The receiver can change its current configuration immediately after receiving a configuration message. The receiver will always use the current configuration only.

The current configuration is loaded from permanent configuration hard-coded in the receiver firmware (the defaults) and from non-volatile memory (user configuration) on startup of the receiver. Changes made to the current configuration at run-time will be lost when there is a power cycle, a hardware reset or a (complete) controlled software reset (see Forcing a Receiver Reset).

See the Configuration Interface section for a detailed description of the receiver configuration system, the explanation of the configuration concept and its principles and interfaces.

ℹ *The configuration interface has changed from earlier u-blox positioning receivers. There is some backwards compatibility. Users are strongly advised to only use the* Configuration Interface *referred to in the following sections. See also* Legacy Configuration Interface Compatibility*.*

3.1 Changing Configuration

All configuration messages, including legacy UBX-CFG messages, will result in an ACK or NACK response. If several configuration messages are sent without waiting for this response then the receiver may pause processing of input messages until processing of a previous configuration message has been completed. When this happens a warning message *wait for cfg ACK* will be sent to the host.

3.2 Basic Receiver Configuration

This section summarises the basic receiver configuration most commonly used.

3.2.1 Communication Interface Configuration

Several configuration items allow operation mode configuration of the various communications interfaces. This includes parameters for the data framing, transfer rate and protocols used. See Serial Communication Ports Description for details. The configuration items available for each interface are:

- UART1 interface: CFG-UART1-*, CFG-UART1INPROT-*, CFG-UART1OUTPROT-*
- UART2 interface: CFG-UART2-*, CFG-UART2INPROT-*, CFG-UART2OUTPROT-*
- SPI interface: CFG-SPI-*, CFG-SPIINPROT-*, CFG-SPIOUTPROT-*
- I2C interface: CFG-I2C-*, CFG-I2CINPROT-*, CFG-I2COUTPROT-*
- USB interface: CFG-USB-*, CFG-USBINPROT-*, CFG-USBOUTPROT-*

ℹ *Not all interfaces are available on all products.*

3.2.2 Message Output Configuration

The rate of NMEA, UBX and RTCM protocol output messages is configurable. If the rate configuration value is zero, then the corresponding message will not be output. Values greater than zero indicate how often the message is output.

For periodic output messages the rate relates to the event the message is related to. For example, the UBX-

NAV-PVT (Navigation Position Velocity and Time Solution) is related to the navigation epoch. If the rate of this message is set to one (1), it will be output for every navigation epoch. If the rate is set to two (2), it will be output every other navigation epoch. The rates of the output messages are individually configurable per communication interface. See the CFG-MSGOUT-* configuration items.

Some messages, such as UBX-MON-VER, are not periodic and will only be output as the answer to a poll request.

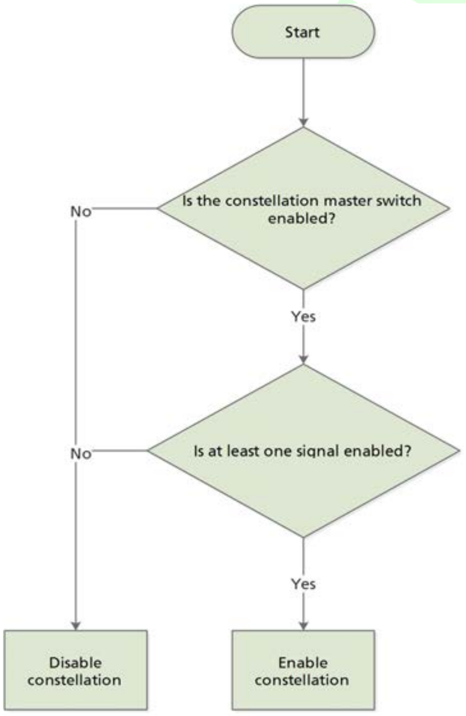
The UBX-INF-* information messages are non-periodic output messages that do not have a message rate configuration. Instead they can be enabled for each communication interface via the CFG-INFMSG-* configuration items.

Note that all message output is additionally subject to the protocol configuration of the communication interfaces. Messages of a given protocol will not be output until the protocol is enabled for output on the interface (see previous section).

3.2.3 GNSS Signal Configuration

Several configuration items govern the use of GNSS signals and constellations. Each GNSS constellation can be controlled separately. Similarly, each signal within the constellation can be handled independently of each other. The constellation enable master keys and their associated signal enable keys are linked such that a GNSS constellation is considered enabled if and only if the master constellation enable key is set and at least one signal associated with that constellation is enabled.

This dependency allows the user to disable a constellation without having to disable each signal on its own. The signal enable keys give users the flexibility to enable or disable each signal separately if they need to. See figure below which describes the flow of enabling / disabling a constellation.


Start
Is the constellation master switch enabled?
No
Yes
Is at least one signal enabled?
No
Yes
Disable constellation
Enable constellation

Flow Diagram of behavior of individual signal switches and master constellation switches

See the table below for an example of possible combinations of values for configuration items related to GPS signals and whether they mean the constellation is enabled or not.

Example of possible values of configuration items for the GPS constellation

master switch	*signal switch*	*signal switch*	*Constellation*
CFG-SIGNAL-GPS_ENA	CFG-SIGNAL-GPS_L1CA_ENA	CFG-SIGNAL-GPS_L2C_ENA	*enabled?*
false (0)	false (0)	false (0)	**no**
false (0)	false (0)	true (1)	**no**
false (0)	true (1)	false (0)	**no**
false (0)	true (1)	true (1)	**no**
true (1)	false (0)	false (0)	**no**
true (1)	false (0)	true (1)	**yes**
true (1)	true (1)	false (0)	**yes**
true (1)	true (1)	true (1)	**yes**

Note that on F9P positioning receivers only some combinations of signals are supported. For all systems both L1 and L2 signals need to be either enabled or disabled, with the exception of the BeiDou B2 signal, which may be disabled individually. Attempts at configuring unsupported combinations will be rejected by the receiver. The receiver responds to such UBX-CFG-VALSET messages with a UBX-ACK-NAK and, if configured, additionally sends the warning *invalid sig cfg*.

3.2.4 Antenna Supervisor Configuration

This section describes the antenna supervisor configuration, its use and restrictions. See the Integration Manual for more information about the hardware and circuits required to make use of the antenna supervisor.

The antenna supervisor is used to control an active antenna. The configuration of the antenna supervisor allows the following:

- Control voltage supply to the antenna, which allows the antenna supervisor to cut power to the antenna at the event of a short circuit or optimise power to the antenna in Power Save Mode
- Detect a short circuit in the antenna and auto recover the antenna supply in such event
- Detect an open antenna, which can be used to tell if the antenna has been disconnected

See the table below, for a description of the configuration items related to the antenna supervisor operation.

Antenna Supervisor Configuration

Configuration Item	*Description*	*Comments*
CFG-HW-ANT_CFG_VOLTCTRL	Enable active antenna voltage control	
CFG-HW-ANT_CFG_SHORTDET	Enable short circuit detection	
CFG-HW-ANT_CFG_OPENDET	Enable open circuit detection	
CFG-HW-ANT_CFG_PWRDOWN	Power Down Antenna supply if Short Circuit is detected	
CFG-HW-ANT_CFG_RECOVER	Enable auto recovery in the event of a short circuit	To use this feature, short circuit detection should be enabled. See CFG-HW-ANT_CFG_SHORTDET
CFG-HW-ANT_SUP_SWITCH_PIN	PIO-Pin used for switching antenna supply	It is recommended that you use the default pins
CFG-HW-ANT_SUP_SHORT_PIN	PIO-Pin used for detecting a short in the antenna supply	It is recommended that you use the default pins
CFG-HW-ANT_SUP_OPEN_PIN	PIO-Pin used for detecting open/not connected antenna	It is recommended that you use the default pins

It is possible to obtain the status of the antenna supervisor through UBX-MON-HW message. Moreover, any changes in the status of the antenna supervisor are reported to the host interface in the form of notice messages. See the tables below for a description of the antenna state status and the antenna power status.

Antenna State Status

Status	*Description*
INIT	Antenna state not defined yet
DONTKNOW	Antenna state is not known
OK	Antenna is ok
SHORT	Antenna short was detected
OPEN	Open antenna was detected

Antenna Power Status

Status	*Description*
OFF	Antenna is off
ON	Antenna is on
DONTKNOW	Antenna power status is not known

3.3 Legacy Configuration Interface Compatibility

There is interface backwards-compatibility for the legacy UBX-CFG configuration messages, such as the UBX-CFG-NAV5 message. See Legacy UBX-CFG Message Fields Reference for details.

The UBX-CFG-CFG interface can be used to save the current configuration into the BBR and Flash layers, clear the current configuration (i.e. the RAM Layer) or can be used to rebuild the RAM Layer from the lower layers.

3.4 Pin Layer Configuration

The Pin Layer holds the following Configuration Items:

- CFG-HW-OSC_TYPE
- CFG-UART1_BAUDRATE

See also Pin Layer Configuration in the Interface Description chapter of this document.

3.5 OTP Layer Configuration

The OTP Layer holds the following Configuration Items from the eFuse Fixed Section:

- CFG-HW-DCDC_DIS
- CFG-HW-SINGLE_CLK
- CFG-HW-OSC_TYPE
- CFG-UART1-REMAP

The OTP Layer holds the following Configuration items from the eFuse Files:

- File 0x30 (see UBX-CFG-OTP-WRITEFILE_30): CFG-UART1-BAUDRATE, CFG-USB-SELFPOW
- File 0x36 (see UBX-CFG-OTP-WRITEFILE_36): CFG-CLOCK-OSC_FREQ, CFG-HW-CLK_OFFSET_VALID, CFG-HW-CLK_OFFSET, CFG-HW-CLK_PRECISION, CFG-HW-CLK_MAX_CALIB_DEV_VALID, CFG-HW-CLK_MAX_CALIB_DEV
- File 0x32 (see UBX-CFG-OTP-WRITEFILE_32): CFG-HW-SINGLE_CLK
- File 0x20 (see UBX-CFG-OTP-WRITEFILE_20): CFG-USB-VENDOR_ID
- File 0x21 (see UBX-CFG-OTP-WRITEFILE_21): CFG-USB-VENDOR_STR0, CFG-USB-VENDOR_STR1,

CFG-USB-VENDOR_STR2, CFG-USB-VENDOR_STR3

- File 0x22 (see UBX-CFG-OTP-WRITEFILE_22): CFG-USB-PRODUCT_ID
- File 0x23 (see UBX-CFG-OTP-WRITEFILE_23): CFG-USB-PRODUCT_STR0, CFG-USB-PRODUCT_STR1, CFG-USB-PRODUCT_STR2, CFG-USB-PRODUCT_STR3

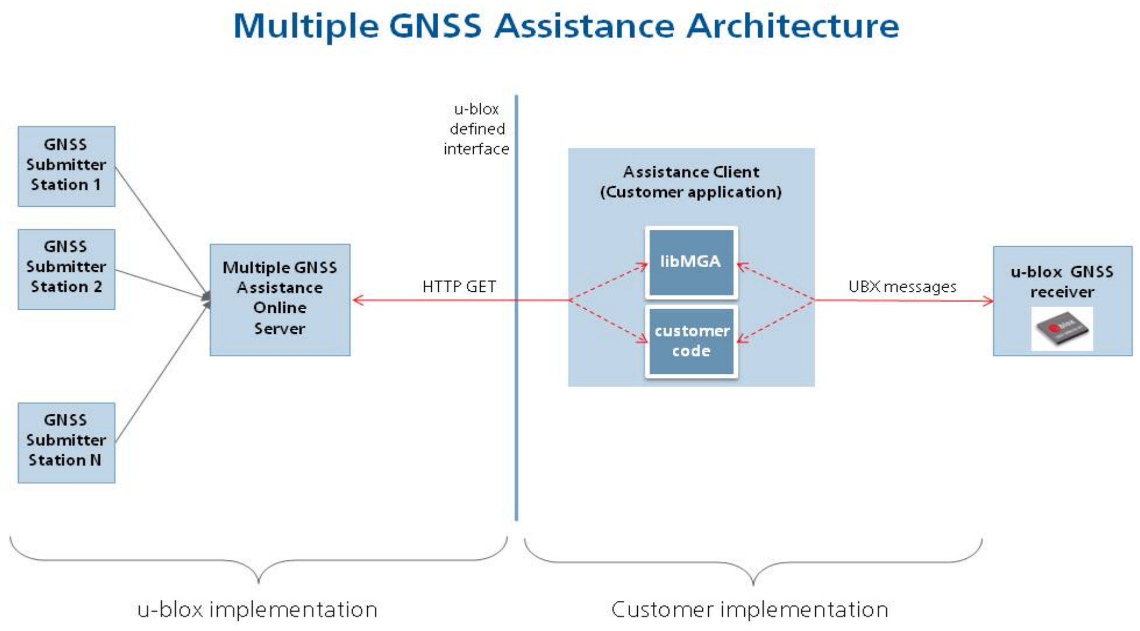
See eFuse for more details on the eFuse OTP memory usage.

4 Multiple GNSS Assistance (MGA)

4.1 AssistNow Online

AssistNow Online is u-blox' end-to-end Assisted GNSS (A-GNSS) solution for receivers that have access to the Internet. Data supplied by the AssistNow Online Service can be directly uploaded to a u-blox receiver in order to substantially reduce Time To First Fix (TTFF), even under poor signal conditions. The system works by collecting data such as ephemeris and almanac from the satellites through u-blox' Global Reference Network of receivers and providing this data to customers in a convenient form that can be forwarded on directly to u-blox receivers.

The AssistNow Online Service uses a simple, stateless, HTTP interface. Therefore, it works on all standard mobile communication networks that support Internet access, including GPRS, UMTS and Wireless LAN. No special arrangements need to be made with mobile network operators to enable AssistNow Online.



The data returned by the AssistNow Online Service is a sequence of UBX-MGA messages, starting with an estimate of the current time in the form of a UBX-MGA-INI-TIME_UTC message.

AssistNow Online currently supports GPS, GLONASS, BeiDou, Galileo, and QZSS.

Customers may choose to use third party sources of assistance data instead of using the AssistNow Online Service. Customers choosing this option will need to ensure that the data is converted from the format used by the third party source to the appropriate MGA messages. However, it is important to ensure that the receiver has an estimate of the current time before it processes any *other assistance data. For this reason, it is strongly recommended to send a `UBX-MGA-INI-TIME_UTC` or `UBX-MGA-INI-TIME_GNSS` as the first message of any assistance.*

4.1.1 Host Software

As u-blox receivers have no means to connect directly with the Internet, the AssistNow Online system can only work if the host system that contains the receiver can connect to the Internet, download the data from the AssistNow Online Service and forward it on to the receiver. In the simplest case that may involve fetching the data from the AssistNow Online Service (by means of a single HTTP GET request), and sending the resulting data to the receiver.

Depending on the circumstances, it may be beneficial for the host software to include:

- Creating an appropriate `UBX-MGA-INI-TIME_UTC` message to deliver a better sense of time to the receiver, especially if the host system has a very good sense of the current time and can deliver a time pulse to one of the receiver's EXTINT pins.
- Enable and use flow control to prevent loss of data due to buffer overflow in the receiver.

u-blox provides the source code for an example library, called libMGA, that provides all of the functionality we expect in most host software.

4.1.2 AssistNow Online Sequence

A typical sequence of use of the AssistNow Online Service comprises the following steps:

- Power-up the u-blox receiver
- Request data from the AssistNow Online Service
- Optionally send `UBX-MGA-INI-TIME_UTC` followed by hardware time synchronization pulse if hardware time synchronization is required.
- Send the UBX messages obtained from the AssistNow Online Service to the receiver.

4.1.3 Flow Control

u-blox receivers aim to process incoming messages as quickly as possible, but there will always be a small delay in processing each message. Uploading assistance data to the receiver can involve sending as many as one hundred of individual messages to the receiver, one after the other. If the communication link is fast, and/or the receiver is busy (trying to acquire new signals), it is possible that the internal buffers will overflow and some messages will be lost. In order to combat this, u-blox receivers support an optional flow control mechanism for assistance.

Flow control is activated by setting the ackAiding parameter in the `UBX-CFG-NAVX5` message. As a result the receiver will issue an acknowledgement message (`UBX-MGA-ACK`) for each assistance message it successfully receives. The host software can examine these acknowledgements to establish whether there were any problems with the data sent to the receiver and deduce (by the lack of acknowledgement) if any messages have been lost. It may then be appropriate to resend some of the assistance messages.

The simplest way to implement flow control would be to send one UBX-MGA assistance message at a time, waiting for the acknowledgement, before sending the next. However, such a strategy is likely to introduce significant delays into the whole assistance process. The best strategy will depend on the amount of assistance data being sent and the nature of the communications link (e.g. baud rate of serial link). u-blox recommends that when customers are developing their host software they start by sending all assistance messages and then analyse the resulting acknowledgements to see whether there have been significant losses. Adding small delays during the transmission may be a simple but effective way to avoid substantial loss of data.

4.1.4 Authorization

The AssistNow Online Service is only available for use by u-blox customers. In order to use the services, customers will need to obtain an authorization token from u-blox. This token must be supplied as a parameter whenever a request is made to either service.

4.1.5 Service Parameters

The information exchange with the AssistNow Online Service is based on the HTTP protocol. Upon reception of an HTTP GET request, the server will respond with the required messages in binary format or with an error string in text format. After delivery of all data, the server will terminate the connection.

The HTTP GET request from the client to the server should contain a standard HTTP query string in the request URL. The query string consists of a set of "key=value" parameters in the following form:

key=value;key=value;key=value;

The following rules apply:

- The order of keys is not important.
- Keys and values are case sensitive.
- Keys and values must be separated by an equals character ('=').
- Key/value pairs must be separated by semicolons (';').
- If a value contains a list, each item in the list must be separated by a comma (',').

The following table describes the keys that are supported.

AssistNow Online Parameter Keys

Key Name	*Unit/Range*	*Optional*	*Description*
token	String	Mandatory	The authorization token supplied by u-blox when a client registers to use the service.
gnss	String	Mandatory	A comma separated list of the GNSS for which data should be returned. Valid GNSS are: gps, qzss and glo.
datatype	String	Mandatory	A comma separated list of the data types required by the client. Valid data types are: eph, alm, aux and pos. Time data is always returned for each request. If the value of this parameter is an empty string, only time data will be returned.
lat	Numeric [degrees]	Optional	Approximate user latitude in WGS 84 expressed in degrees and fractional degrees. Must be in range -90 to 90. Example: lat=47.2.
lon	Numeric [degrees]	Optional	Approximate user longitude in WGS 84 expressed in degrees and fractional degrees. Must be in range -180 to 180. Example: lon=8.55.
alt	Numeric [meters]	Optional	Approximate user altitude above WGS 84 Ellipsoid. If this value is not provided, the server assumes an altitude of 0 meters. Must be in range -1000 to 50000.
pacc	Numeric [meters]	Optional	Approximate accuracy of submitted position (see position parameters note below). If this value is not provided, the server assumes an accuracy of 300km. Must be in range 0 to 6000000.
tacc	Numeric [seconds]	Optional	The timing accuracy (see time parameters note below). If this value is not provided, the server assumes an accuracy of 10 seconds. Must be in range 0 to 3600.

AssistNow Online Parameter Keys continued

Key Name	*Unit/Range*	*Optional*	*Description*
latency	Numeric [seconds]	Optional	Typical latency between the time the server receives the request, and the time when the assistance data arrives at the u-blox receiver. The server can use this value to correct the time being transmitted to the client. If this value is not provided, the server assumes a latency of 0. Must be in range 0 to 3600.
filteronpos	(no value required)	Optional	If present, the ephemeris data returned to the client will only contain data for the satellites which are likely to be visible from the approximate position provided by the lat, lon, alt and pacc parameters. If the lat and lon parameters are not provided the service will return an error.
filteronsv	String	Optional	A comma separated list of u-blox gnssId:svId pairs. The ephemeris data returned to the client will only contain data for the listed satellites.

Thus, as an example, a valid parameter string would be:

token=XXXXXXXXXXXXXXXXXXXXXX;gnss=gps,qzss;datatype=eph,pos,aux;lat=47.28;lon=8.56;pacc=1000

4.1.5.1 Position parameters (lat, lon, alt and pacc)

The position parameters (lat, lon, alt and pacc) are used by the server for two purposes:

- If the filteronpos parameter is provided, the server determines the currently visible satellites at the user position, and only sends the ephemeris data of those satellites which should be in view at the location of the user. This reduces bandwidth requirements. In this case the 'pacc' value is taken into account, meaning that the server will return all SVs visible in the given uncertainty region.
- If the datatype 'pos' is requested, the server will return the position and accuracy in the response data. When this data is supplied to the u-blox receiver, depending on the accuracy of the provided data, the receiver can then choose to select a better startup strategy. For example, if the position is accurate to 100km or better, the u-blox receiver will choose to go for a more optimistic startup strategy. This will result in quicker startup time. The receiver will decide which strategy to choose, depending on the 'pacc' parameter. If the submitted user position is less accurate than what is being specified with the 'pacc' parameter, then the user will experience prolonged or even failed startups.

4.1.5.2 Time parameters (tacc and latency)

Time data is always returned with each request. The time data refers to the time at which the response leaves the server, corrected by an optional latency value.This time data provided by the service is accurate to approximately 10ms but by default the time accuracy is indicated to be +/-10 seconds in order to account for network latency and any time between the client receiving the data and it being provided to the receiver.

If both the network latency and the client latency can safely be assumed to be very low (or are known), the client can choose to set the accuracy of the time message (tacc) to a much smaller value (e.g. 0.5s). This will result in a faster TTFF. The latency can also be adjusted as appropriate. However, these fields should be used with caution: if the time accuracy is not correct when the time data reaches the receiver, the receiver may experience prolonged or even failed start-ups.

For optimal results, the client should establish an accurate sense of time itself (e.g. by calibrating its system clock using a local NTP service) and then modify the time data received from the service as appropriate.

4.1.6 Multiple Servers

u-blox has designed and implemented the AssistNow Online Service in a way that should provide very high reliability. Nonetheless, there will be rare occasions when a server is not available (e.g. due to failure or some form of maintenance activity). In order to protect customers against the impact of such outages, u-blox will run at least two instances of the AssistNow Online Service on independent machines. Customers will have a free choice of requesting assistance data from any of these servers, as all will provide the same information. However, should one fail for whatever reason, it is highly unlikely that the other server(s) will also be unavailable. Therefore customers requiring the best possible availability are recommended to implement a scheme where they direct their requests to a chosen server, but, if that server fails to respond, have a fall-back mechanism to use another server instead.

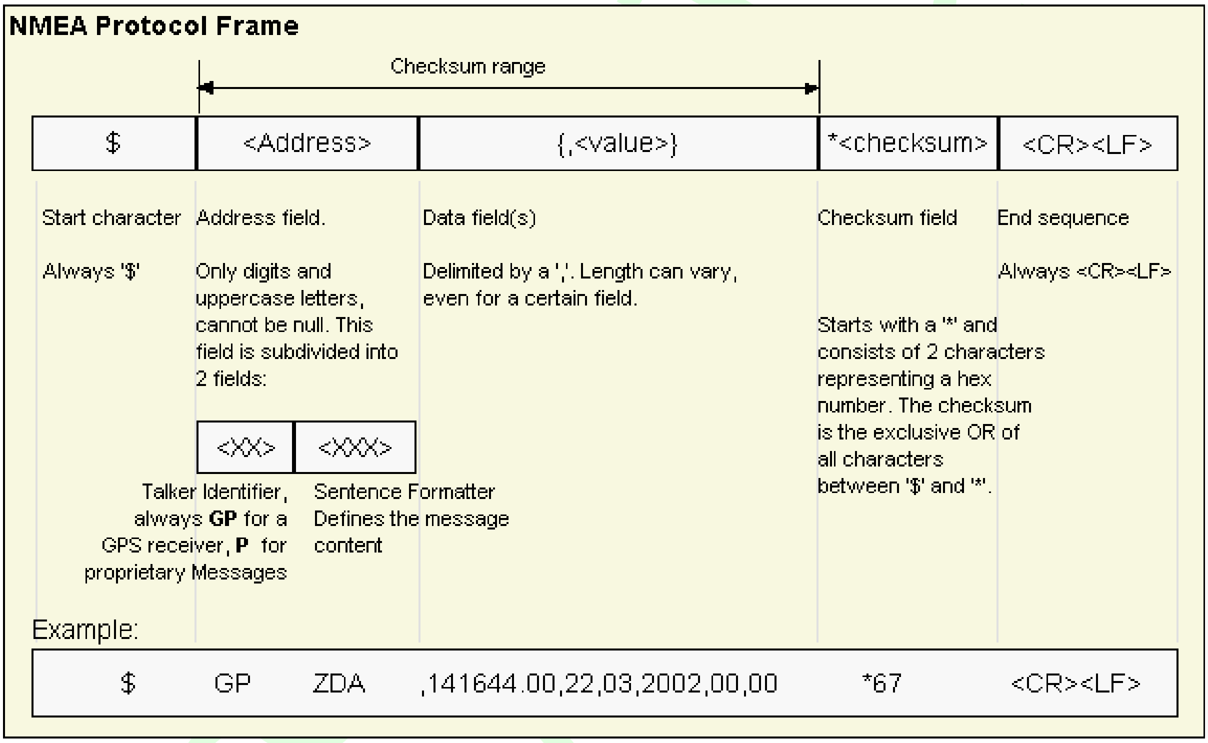
Interface Description

5 NMEA Protocol

5.1 Protocol Overview

5.1.1 Message Format

NMEA messages sent by the GNSS receiver are based on NMEA 0183 Version 4.1. The following picture shows the structure of a NMEA protocol message.



Start character

Always '$'

Address field.

Only digits and uppercase letters, cannot be null. This field is subdivided into 2 fields:

Talker Identifier, always **GP** for a GPS receiver, **P** for proprietary Messages

Sentence Formatter Defines the message content

Data field(s)

Delimited by a ','. Length can vary, even for a certain field.

Checksum field

Starts with a '*' and consists of 2 characters representing a hex number. The checksum is the exclusive OR of all characters between '$' and '*'.

End sequence

Always <CR><LF>

```
$    GP    ZDA    ,141644.00,22,03,2002,00,00    *67    <CR><LF>
```

For further information on the NMEA Standard, refer to *NMEA 0183 Standard For Interfacing Marine Electronic Devices*, Version 4.10, June, 2012. See http://www.nmea.org/ for ordering instructions.

The NMEA standard allows for proprietary, manufacturer-specific messages to be added. These shall be marked with a manufacturer mnemonic. The mnemonic assigned to u-blox is UBX and is used for all non-standard messages. These proprietary NMEA messages therefore have the address field set to PUBX. The first data field in a PUBX message identifies the message number with two digits.

5.1.2 Talker ID

One of the ways the NMEA standard differentiates between GNSS is by using a two-letter message identifier, the 'Talker ID'. The specific Talker ID used by a u-blox receiver will depend on the device model and system configuration. The table below shows the Talker ID that will be used for various GNSS configurations.

NMEA Talker IDs

Configured GNSS	*Talker ID*
GPS, SBAS, QZSS	GP
GLONASS	GL
Galileo	GA

NMEA Talker IDs continued

Configured GNSS	*Talker ID*
BeiDou	GB
Any combination of GNSS	GN

5.1.3 Protocol Configuration

The NMEA protocol on u-blox receivers can be configured to the need of customer applications using `UBX-CFG-NMEA`. For backwards compatibility various versions of this message are supported, however, any new users should use the version that is not marked as deprecated.

There are four NMEA standards supported. The default NMEA version is 4.10. Alternatively versions 4.00, 2.3, and 2.1 can be enabled (for details on how this affects the output refer to section Position Fix Flags in NMEA Mode).

NMEA defines satellite numbering systems for some, but not all GNSS (this is partly dependent on the NMEA version). Satellite numbers for unsupported GNSS can be configured using `UBX-CFG-NMEA`. Unknown satellite numbers are always reported as a null NMEA field (i.e. an empty string)

The NMEA specification indicates that the GGA message is GPS specific. However, u-blox receivers support the output of a GGA message for each of the Talker IDs.

NMEA filtering flags

Parameter	*Description*
Position filtering	Enable to permit positions from failed or invalid fixes to be reported (with the "V" status flag to indicate that the data is not valid).
Valid position filtering	Enable to permit positions from invalid fixes to be reported (with the "V" status flag to indicate that the data is not valid).
Time filtering	Enable to permit the receiver's best knowledge of time to be output, even though it might be wrong.
Date filtering	Enable to permit the receiver's best knowledge of date to be output, even though it might be wrong.
GPS-only filtering	Enable to restrict output to only report GPS satellites.
Track filtering	Enable to permit course over ground (COG) to be reported even when it would otherwise be frozen.

NMEA flags

Parameter	*Description*
Compatibility Mode	Some older NMEA applications expect the NMEA output to be formatted in a specific way, for example, they will only work if the latitude and longitude have exactly four digits behind the decimal point. u-blox receivers offer a compatibility mode to support these legacy applications.
Consideration Mode	u-blox receivers use a sophisticated signal quality detection scheme, in order to produce the best possible position output. This algorithm considers all SV measurements, and may eventually decide to only use a subset thereof, if it improves the overall position accuracy. If Consideration mode is enabled, all satellites, which were considered for navigation, are communicated as being used for the position determination. If Consideration Mode is disabled, only those satellites which after the consideration step remained in the position output are marked as being used.
Limit82 Mode	Enabling this mode will limit the NMEA sentence length to a maximum of 82 characters.

NMEA flags continued

Parameter	*Description*
High Precision Mode	Enabling this mode increases precision of the position output. Latitude and longitude then have seven digits after the decimal point, and altitude has three digits after the decimal point. Note: The High Precision Mode cannot be set in conjunction with either Compatibility Mode or Limit82 Mode.

Extended configuration

Option	*Description*
GNSS to filter	Filters satellites based on their GNSS
Satellite numbering	This field configures the display of satellites that do not have an NMEA-defined value. Note: this does not apply to satellites with an unknown ID.
Main Talker ID	By default the main Talker ID (i.e. the Talker ID used for all messages other than GSV) is determined by the GNSS assignment of the receiver's channels (see `UBX-CFG-GNSS`). This field enables the main Talker ID to be overridden.
GSV Talker ID	By default the Talker ID for GSV messages is GNSS specific (as defined by NMEA). This field enables the GSV Talker ID to be overridden.
BDS Talker ID	By default the Talker ID for BeiDou is 'GB'. This field enables the BeiDou Talker ID to be overridden.

Extra fields in NMEA 4.1 and above

Message	*Extra fields*
GBS	systemId, signalId
GNS	navStatus
GRS	systemId, signalId
GSA	systemId
GSV	signalId
RMC	navStatus

5.1.4 Satellite Numbering

The NMEA protocol (V4.1) identifies GNSS satellites with a one digit system ID and a two digit satellite number. u-blox receivers support this method in their NMEA output when "strict" SV numbering is selected. In most cases this is the default setting, but can be checked or set using `UBX-CFG-NMEA`.

In order to support QZSS within current receivers and prepare for support of other systems (e.g. Galileo) in future receivers, an "extended" SV numbering scheme can be enabled (using `UBX-CFG-NMEA`). This uses the NMEA-defined numbers where possible, but adds other number ranges to support other GNSS. Note however that these non-standard extensions require 3 digit numbers, which may not be supported by some NMEA parsing software. For example QZSS satellites are reported using numbers in the range 193 to 197.

See Satellite Numbering for a complete list of satellite numbers.

GLONASS satellites can be tracked before they have been identified. In NMEA output, such unknown satellite numbers are always reported as a null field (i.e. an empty string).

5.1.5 Latitude and Longitude Format

According to the NMEA Standard, Latitude and Longitude are output in the format Degrees, Minutes and (Decimal) Fractions of Minutes. To convert to Degrees and Fractions of Degrees, or Degrees, Minutes, Seconds and Fractions of seconds, the 'Minutes' and 'Fractional Minutes' parts need to be converted. In other words: If the GPS Receiver reports a Latitude of 4717.112671 North and Longitude of 00833.914843 East, this is

Latitude 47 Degrees, 17.112671 Minutes

Longitude 8 Degrees, 33.914843 Minutes

or

Latitude 47 Degrees, 17 Minutes, 6.76026 Seconds

Longitude 8 Degrees, 33 Minutes, 54.89058 Seconds

or

Latitude 47.28521118 Degrees

Longitude 8.56524738 Degrees

5.1.6 Position Fix Flags

This section shows how u-blox implements the NMEA protocol and the conditions determining how flags are set.

Flags in NMEA 4.1 and above

NMEA Message	*GLL, RMC*	*GGA*	*GLL, VTG*	*RMC, GNS*
Field	*status*	*quality*	*posMode*	*posMode*
No position fix (at power-up, after losing satellite lock)	V	0	N	N
GNSS fix, but user limits exceeded	V	0	N	N
Dead reckoning fix, but user limits exceeded	V	6	E	E
Dead reckoning fix	A	6	E	E
RTK float	A	5	D	F
RTK fixed	A	4	D	R
2D GNSS fix	A	1 / 2	A / D	A / D
3D GNSS fix	A	1 / 2	A / D	A / D
Combined GNSS/dead reckoning fix	A	1 / 2	A / D	A / D
	See below (1)	See below (2)	See below (3)	See below (3)

(1) Possible values for *status*: V = Data invalid, A = Data valid

(2) Possible values for *quality*: 0 = No fix, 1 = Autonomous GNSS fix, 2 = Differential GNSS fix, 4 = RTK fixed, 5 = RTK float, 6 = Estimated/Dead reckoning fix

(3) Possible values for *posMode*: N = No fix, E = Estimated/Dead reckoning fix, A = Autonomous GNSS fix, D = Differential GNSS fix, F = RTK float, R = RTK fixed

Flags in NMEA 2.3 and above

NMEA Message	*GLL, RMC*	*GGA*	*GSA*	*GLL, VTG, RMC, GNS*
Field	*status*	*quality*	*navMode*	*posMode*
No position fix (at power-up, after losing satellite lock)	V	0	1	N
GNSS fix, but user limits exceeded	V	0	1	N
Dead reckoning fix, but user limits exceeded	V	6	2	E
Dead reckoning fix	A	6	2	E
2D GNSS fix	A	1 / 2	2	A / D
3D GNSS fix	A	1 / 2	3	A / D
Combined GNSS/dead reckoning fix	A	1 / 2	3	A / D
	See below (1)	See below (2)	See below (3)	See below (4)

(1) Possible values for *status*: V = Data invalid, A = Data valid

(2) Possible values for *quality*: 0 = No fix, 1 = Autonomous GNSS fix, 2 = Differential GNSS fix, 4 = RTK fixed, 5 = RTK float, 6 = Estimated/Dead reckoning fix

(3) Possible values for *navMode*: 1 = No fix, 2 = 2D fix, 3 = 3D fix

(4) Possible values for *posMode*: N = No fix, E = Estimated/Dead reckoning fix, A = Autonomous GNSS fix, D = Differential GNSS fix, F = RTK float, R = RTK fixed

Flags in NMEA 2.1 and below

The flags in NMEA 2.1 and below are the same as NMEA 2.3 and above but with the following differences:

- The *posMode* field is not output for GLL, RMC and VTG messages (each message has one field less).
- The GGA *quality* field is set to 1 (instead of 6) for both types of dead reckoning fix.

5.1.7 Multi-GNSS Considerations

Many applications which process NMEA messages assume that only a single GNSS is active. However, when multiple GNSS are configured, the NMEA specification requires the output to change in the following ways:

NMEA output for Multi-GNSS

Change	*Description*
Main Talker ID	The main Talker ID will be 'GN' (e.g. instead of 'GP' for a GPS receiver)
GSV Talker IDs	The GSV message reports the signal strength of the visible satellites. However, the Talker ID it uses is specific to the GNSS it is reporting information for, so for a multi-GNSS receiver it will not be the same as the main Talker ID. (e.g. other messages will be using the 'GN' Talker ID but the GSV message will use GNSS-specific Talker IDs)
Multiple GSA and GRS Messages	Multiple GSA and GRS messages are output for each fix, one for each GNSS. This may confuse applications which assume they are output only once per position fix (as is the case for a single GNSS receiver).

5.1.8 Output of Invalid/Unknown Data

By default the receiver will not output invalid data. In such cases, it will output empty fields.

A valid position fix is reported as follows:

```
$GPGLL,4717.11634,N,00833.91297,E,124923.00,A,A*6E
```

An invalid position fix (but time valid) is reported as follows:

```
$GPGLL,,,,,124924.00,V,N*42
```

If Time is unknown (e.g. during a cold-start):

```
$GPGLL,,,,,,V,N*64
```

Note:

> *Output of invalid data marked with the 'Invalid/Valid' Flags can be enabled using the UBX protocol message* UBX-CFG-NMEA.

5.1.9 Messages Overview

When configuring NMEA messages using the UBX protocol message UBX-CFG-MSG, the Class/Ids shown in the table shall be used.

Page	*Mnemonic*	*Cls/ID*	*Description*
	NMEA Standard Messages		**Standard Messages**
21	**DTM**	0xF0 0x0A	Datum Reference

NMEA Messages Overview continued

Page	*Mnemonic*	*Cls/ID*	*Description*
22	**GBQ**	0xF0 0x44	Poll a standard message (if the current Talker ID is GB)
22	**GBS**	0xF0 0x09	GNSS Satellite Fault Detection
23	**GGA**	0xF0 0x00	Global positioning system fix data
25	**GLL**	0xF0 0x01	Latitude and longitude, with time of position fix and status
26	**GLQ**	0xF0 0x43	Poll a standard message (if the current Talker ID is GL)
26	**GNQ**	0xF0 0x42	Poll a standard message (if the current Talker ID is GN)
27	**GNS**	0xF0 0x0D	GNSS fix data
28	**GPQ**	0xF0 0x40	Poll a standard message (if the current Talker ID is GP)
28	**GRS**	0xF0 0x06	GNSS Range Residuals
29	**GSA**	0xF0 0x02	GNSS DOP and Active Satellites
30	**GST**	0xF0 0x07	GNSS Pseudo Range Error Statistics
31	**GSV**	0xF0 0x03	GNSS Satellites in View
32	**RMC**	0xF0 0x04	Recommended Minimum data
33	**TXT**	0xF0 0x41	Text Transmission
34	**VLW**	0xF0 0x0F	Dual ground/water distance
35	**VTG**	0xF0 0x05	Course over ground and Ground speed
36	**ZDA**	0xF0 0x08	Time and Date

5.2 Standard Messages

Standard Messages: i.e. Messages as defined in the NMEA Standard.

5.2.1 DTM

5.2.1.1 Datum Reference

Message	**DTM**		
Description	**Datum Reference**		
Firmware	Supported on: • u-blox 9 with protocol version 27		
Type	Output Message		
Comment	This message gives the difference between the current datum and the reference datum. The current datum defaults to WGS84 The reference datum cannot be changed and is always set to WGS84.		
	ID for CFG-MSG	*Number of fields*	
Message Info	0xF0 0x0A	11	

Message Structure:

```
$xxDTM,datum,subDatum,lat,NS,lon,EW,alt,refDatum*cs<CR><LF>
```

Example:

```
$GPDTM,W84,,0.0,N,0.0,E,0.0,W84*6F
$GPDTM,999,,0.08,N,0.07,E,-47.7,W84*1C
```

Field No.	*Name*	*Unit*	*Format*	*Example*	*Description*
0	`xxDTM`	-	string	$GPDTM	DTM Message ID (xx = current Talker ID)
1	`datum`	-	string	W84	Local datum code: W84 = WGS84, 999 = user defined
2	`subDatum`	-	string	-	A null field
3	`lat`	min	numeric	0.08	Offset in Latitude
4	`NS`	-	character	S	North/South indicator
5	`lon`	min	numeric	0.07	Offset in Longitude
6	`EW`	-	character	E	East/West indicator
7	`alt`	m	numeric	-2.8	Offset in altitude
8	`refDatum`	-	string	W84	Reference datum code (always W84 = WGS 84)
9	`cs`	-	hexadecimal	*67	Checksum
10	`<CR><LF>`	-	character	-	Carriage return and line feed

5.2.2 GBQ

5.2.2.1 Poll a standard message (if the current Talker ID is GB)

Message	**GBQ**		
Description	**Poll a standard message (if the current Talker ID is GB)**		
Firmware	Supported on: • u-blox 9 with protocol version 27		
Type	Input Message		
Comment	Polls a standard NMEA message if the current Talker ID is GB		
Message Info	*ID for CFG-MSG*	*Number of fields*	
	0xF0 0x44	4	

Message Structure:

```
$xxGBQ,msgId*cs<CR><LF>
```

Example:

```
$EIGBQ,RMC*28
```

Field No.	*Name*	*Unit*	*Format*	*Example*	*Description*
0	`xxGBQ`	-	string	$EIGBQ	GBQ Message ID (xx = Talker ID of the device requesting the poll)
1	`msgId`	-	string	RMC	Message ID of the message to be polled
2	`cs`	-	hexadecimal	*28	Checksum
3	`<CR><LF>`	-	character	-	Carriage return and line feed

5.2.3 GBS

5.2.3.1 GNSS Satellite Fault Detection

Message	**GBS**		
Description	**GNSS Satellite Fault Detection**		
Firmware	Supported on: • u-blox 9 with protocol version 27		
Type	Output Message		
Comment	This message outputs the results of the Receiver Autonomous Integrity Monitoring Algorithm (RAIM). • The fields **errLat**, **errLon** and **errAlt** output the standard deviation of the position calculation, using all satellites which pass the RAIM test successfully. • The fields **errLat**, **errLon** and **errAlt** are only output if the RAIM process passed successfully (i.e. no or successful edits happened). These fields are never output if 4 or fewer satellites are used for the navigation calculation (because, in such cases, integrity can not be determined by the receiver autonomously). • The fields **prob**, **bias** and **stdev** are only output if at least one satellite failed in the RAIM test. If more than one satellites fail the RAIM test, only the information for the worst satellite is output in this message.		
Message Info	*ID for CFG-MSG*	*Number of fields*	
	0xF0 0x09	13	

Message Structure:

```
$xxGBS,time,errLat,errLon,errAlt,svid,prob,bias,stddev,systemId,signalId*cs<CR><LF>
```

Example:

```
$GPGBS,235503.00,1.6,1.4,3.2,,,,,,*40
$GPGBS,235458.00,1.4,1.3,3.1,03,,-21.4,3.8,1,0*5B
```

Field No.	*Name*	*Unit*	*Format*	*Example*	*Description*
0	`xxGBS`	-	string	$GPGBS	GBS Message ID (xx = current Talker ID)
1	`time`	-	hhmmss.ss	235503.00	UTC time to which this RAIM sentence belongs, see note on UTC representation
2	`errLat`	m	numeric	1.6	Expected error in latitude
3	`errLon`	m	numeric	1.4	Expected error in longitude
4	`errAlt`	m	numeric	3.2	Expected error in altitude
5	`svid`	-	numeric	03	Satellite ID of most likely failed satellite
6	`prob`	-	numeric	-	Probability of missed detection, not supported (empty)
7	`bias`	m	numeric	-21.4	Estimate on most likely failed satellite (a priori residual)
8	`stddev`	m	numeric	3.8	Standard deviation of estimated bias
9	`systemId`	-	numeric	1	NMEA defined GNSS System ID **NMEA v4.1 and above only**
10	`signalId`	-	numeric	0	NMEA defined GNSS Signal ID (0 = All signals, see Signal Identifiers table for other values) **NMEA v4.1 and above only**
11	`cs`	-	hexadecimal	*5B	Checksum
12	`<CR><LF>`	-	character	-	Carriage return and line feed

5.2.4 GGA

5.2.4.1 Global positioning system fix data

Message	**GGA**		
Description	**Global positioning system fix data**		
Firmware	Supported on: • u-blox 9 with protocol version 27		
Type	Output Message		
Comment	**The output of this message is dependent on the currently selected datum (default: WGS84). The NMEA specification indicates that the GGA message is GPS specific. However, when the receiver is configured for multi-GNSS, the GGA message contents will be generated from the multi-GNSS solution. For multi-GNSS use, it is recommended that the `NMEA-GNS` message is used instead.** Time and position, together with GPS fixing related data (number of satellites in use, and the resulting HDOP, age of differential data if in use, etc.).		
	ID for CFG-MSG	*Number of fields*	
Message Info	0xF0 0x00	17	

Message Structure:

```
$xxGGA,time,lat,NS,long,EW,quality,numSV,HDOP,alt,M,sep,M,diffAge,diffStation*cs<CR><LF>
```

Example:

```
$GPGGA,092725.00,4717.11399,N,00833.91590,E,1,08,1.01,499.6,M,48.0,M,,*5B
```

GGA continued

Field No.	*Name*	*Unit*	*Format*	*Example*	*Description*
Field No.	*Name*	*Unit*	*Format*	*Example*	*Description*
0	`xxGGA`	-	string	$GPGGA	GGA Message ID (xx = current Talker ID)
1	`time`	-	hhmmss.ss	092725.00	UTC time, see note on UTC representation
2	`lat`	-	ddmm. mmmmm	4717.11399	Latitude (degrees & minutes), see format description
3	`NS`	-	character	N	North/South indicator
4	`long`	-	dddmm. mmmmm	00833.91590	Longitude (degrees & minutes), see format description
5	`EW`	-	character	E	East/West indicator
6	`quality`	-	digit	1	Quality indicator for position fix, see table below and position fix flags description
7	`numSV`	-	numeric	08	Number of satellites used (range: 0-12)
8	`HDOP`	-	numeric	1.01	Horizontal Dilution of Precision
9	`alt`	m	numeric	499.6	Altitude above mean sea level
10	`uAlt`	-	character	M	Altitude units: meters (fixed field)
11	`sep`	m	numeric	48.0	Geoid separation: difference between ellipsoid and mean sea level
12	`uSep`	-	character	M	Separation units: meters (fixed field)
13	`diffAge`	s	numeric	-	Age of differential corrections (blank when DGPS is not used)
14	`diffStation`	-	numeric	-	ID of station providing differential corrections (blank when DGPS is not used)
15	`cs`	-	hexadecimal	*5B	Checksum
16	`<CR><LF>`	-	character	-	Carriage return and line feed

Table Quality Indicator

Quality Indicator	*Description, see also position fix flags description*
0	No Fix / Invalid
1	Standard GPS (2D/3D)
2	Differential GPS
4	RTK fixed solution
5	RTK float solution
6	Estimated (DR) Fix

5.2.5 GLL

5.2.5.1 Latitude and longitude, with time of position fix and status

Message	**GLL**		
Description	**Latitude and longitude, with time of position fix and status**		
Firmware	Supported on: • u-blox 9 with protocol version 27		
Type	Output Message		
Comment	**The output of this message is dependent on the currently selected datum (default: WGS84)** -		
Message Info	*ID for CFG-MSG*	*Number of fields*	
	0xF0 0x01	10	

Message Structure:

```
$xxGLL,lat,NS,long,EW,time,status,posMode*cs<CR><LF>
```

Example:

```
$GPGLL,4717.11364,N,00833.91565,E,092321.00,A,A*60
```

Field No.	*Name*	*Unit*	*Format*	*Example*	*Description*
0	`xxGLL`	-	string	$GPGLL	GLL Message ID (xx = current Talker ID)
1	`lat`	-	ddmm. mmmmm	4717.11364	Latitude (degrees & minutes), see format description
2	`NS`	-	character	N	North/South indicator
3	`long`	-	dddmm. mmmmm	00833.91565	Longitude (degrees & minutes), see format description
4	`EW`	-	character	E	East/West indicator
5	`time`	-	hhmmss.ss	092321.00	UTC time, see note on UTC representation
6	`status`	-	character	A	V = Data invalid or receiver warning, A = Data valid. See position fix flags description.
7	`posMode`	-	character	A	Positioning mode, see position fix flags description. **NMEA v2.3 and above only**
8	`cs`	-	hexadecimal	*60	Checksum
9	`<CR><LF>`	-	character	-	Carriage return and line feed

5.2.6 GLQ

5.2.6.1 Poll a standard message (if the current Talker ID is GL)

Message	**GLQ**		
Description	**Poll a standard message (if the current Talker ID is GL)**		
Firmware	Supported on: • u-blox 9 with protocol version 27		
Type	Input Message		
Comment	Polls a standard NMEA message if the current Talker ID is GL		
Message Info	*ID for CFG-MSG*	*Number of fields*	
	0xF0 0x43	4	

Message Structure:

```
$xxGLQ,msgId*cs<CR><LF>
```

Example:

```
$EIGLQ,RMC*3A
```

Field No.	*Name*	*Unit*	*Format*	*Example*	*Description*
0	xxGLQ	-	string	$EIGLQ	GLQ Message ID (xx = Talker ID of the device requesting the poll)
1	msgId	-	string	RMC	Message ID of the message to be polled
2	cs	-	hexadecimal	*3A	Checksum
3	<CR><LF>	-	character	-	Carriage return and line feed

5.2.7 GNQ

5.2.7.1 Poll a standard message (if the current Talker ID is GN)

Message	**GNQ**		
Description	**Poll a standard message (if the current Talker ID is GN)**		
Firmware	Supported on: • u-blox 9 with protocol version 27		
Type	Input Message		
Comment	Polls a standard NMEA message if the current Talker ID is GN		
Message Info	*ID for CFG-MSG*	*Number of fields*	
	0xF0 0x42	4	

Message Structure:

```
$xxGNQ,msgId*cs<CR><LF>
```

Example:

```
$EIGNQ,RMC*3A
```

Field No.	*Name*	*Unit*	*Format*	*Example*	*Description*
0	xxGNQ	-	string	$EIGNQ	GNQ Message ID (xx = Talker ID of the device requesting the poll)
1	msgId	-	string	RMC	Message ID of the message to be polled
2	cs	-	hexadecimal	*3A	Checksum
3	<CR><LF>	-	character	-	Carriage return and line feed

5.2.8 GNS

5.2.8.1 GNSS fix data

Message	**GNS**		
Description	**GNSS fix data**		
Firmware	Supported on: • u-blox 9 with protocol version 27		
Type	Output Message		
Comment	**The output of this message is dependent on the currently selected datum (default: WGS84)** Time and position, together with GNSS fixing related data (number of satellites in use, and the resulting HDOP, age of differential data if in use, etc.).		
Message Info	*ID for CFG-MSG*	*Number of fields*	
	0xF0 0x0D	16	

Message Structure:

```
$xxGNS,time,lat,NS,long,EW,posMode,numSV,HDOP,alt,altRef,diffAge,diffStation,navStatus*cs<CR><LF>
```

Example:

```
$GPGNS,091547.00,5114.50897,N,00012.28663,W,AA,10,0.83,111.1,45.6,,,V*71
```

Field No.	*Name*	*Unit*	*Format*	*Example*	*Description*
0	`xxGNS`	-	string	$GPGNS	GNS Message ID (xx = current Talker ID)
1	`time`	-	hhmmss.ss	091547.00	UTC time, see note on UTC representation
2	`lat`	-	ddmm. mmmmm	5114.50897	Latitude (degrees & minutes), see format description
3	`NS`	-	character	N	North/South indicator
4	`long`	-	dddmm. mmmmm	00012.28663	Longitude (degrees & minutes), see format description
5	`EW`	-	character	E	East/West indicator
6	`posMode`	-	character	AA	Positioning mode, see position fix flags description. First character for GPS, second character for GLONASS
7	`numSV`	-	numeric	10	Number of satellites used (range: 0-99)
8	`HDOP`	-	numeric	0.83	Horizontal Dilution of Precision
9	`alt`	m	numeric	111.1	Altitude above mean sea level
10	`sep`	m	numeric	45.6	Geoid separation: difference between ellipsoid and mean sea level
11	`diffAge`	s	numeric	-	Age of differential corrections (blank when DGPS is not used)
12	`diffStation`	-	numeric	-	ID of station providing differential corrections (blank when DGPS is not used)
13	`navStatus`	-	character	V	Navigational status indicator (V = Equipment is not providing navigational status information) **NMEA v4.1 and above only**
14	`cs`	-	hexadecimal	*71	Checksum
15	`<CR><LF>`	-	character	-	Carriage return and line feed

5.2.9 GPQ

5.2.9.1 Poll a standard message (if the current Talker ID is GP)

Message	**GPQ**		
Description	**Poll a standard message (if the current Talker ID is GP)**		
Firmware	Supported on: - u-blox 9 with protocol version 27		
Type	Input Message		
Comment	Polls a standard NMEA message if the current Talker ID is GP		
	ID for CFG-MSG	*Number of fields*	
Message Info	0xF0 0x40	4	

Message Structure:

```
$xxGPQ,msgId*cs<CR><LF>
```

Example:

```
$EIGPQ,RMC*3A
```

Field No.	*Name*	*Unit*	*Format*	*Example*	*Description*
0	`xxGPQ`	-	string	$EIGPQ	GPQ Message ID (xx = Talker ID of the device requesting the poll)
1	`msgId`	-	string	RMC	Message ID of the message to be polled
2	`cs`	-	hexadecimal	*3A	Checksum
3	`<CR><LF>`	-	character	-	Carriage return and line feed

5.2.10 GRS

5.2.10.1 GNSS Range Residuals

Message	**GRS**		
Description	**GNSS Range Residuals**		
Firmware	Supported on: - u-blox 9 with protocol version 27		
Type	Output Message		
Comment	**This messages relates to associated GGA and GSA messages.** If less than 12 SVs are available, the remaining fields are output empty. If more than 12 SVs are used, only the residuals of the first 12 SVs are output, in order to remain consistent with the NMEA standard. **In a multi-GNSS system this message will be output multiple times, once for each GNSS.**		
	ID for CFG-MSG	*Number of fields*	
Message Info	0xF0 0x06	19	

Message Structure:

```
$xxGRS,time, mode {,residual},systemId,signalId*cs<CR><LF>
```

Example:

```
$GPGRS,082632.00,1,0.54,0.83,1.00,1.02,-2.12,2.64,-0.71,-1.18,0.25,,,1,0*70
```

Field No.	*Name*	*Unit*	*Format*	*Example*	*Description*
0	`xxGRS`	-	string	$GPGRS	GRS Message ID (xx = current Talker ID)

GRS continued

Field No.	*Name*	*Unit*	*Format*	*Example*	*Description*
1	`time`	-	hhmmss.ss	082632.00	UTC time of associated position fix, see note on UTC representation
2	`mode`	-	digit	1	Mode (see table below), u-blox receivers will always output Mode 1 residuals
Start of repeated block (12 times)					
3 + 1*N	`residual`	m	numeric	0.54	Range residuals for SVs used in navigation. The SV order matches the order from the GSA sentence.
End of repeated block					
15	`systemId`	-	numeric	1	NMEA defined GNSS System ID **NMEA v4.1 and above only**
16	`signalId`	-	numeric	0	NMEA defined GNSS Signal ID (0 = All signals, see Signal Identifiers table for other values) **NMEA v4.1 and above only**
17	`cs`	-	hexadecimal	*70	Checksum
18	`<CR><LF>`	-	character	-	Carriage return and line feed

Table Mode

Mode	*Description*
0	Residuals were used to calculate the position given in the matching GGA sentence.
1	Residuals were recomputed after the GGA position was computed.

5.2.11 GSA

5.2.11.1 GNSS DOP and Active Satellites

Message	**GSA**	
Description	**GNSS DOP and Active Satellites**	
Firmware	Supported on: • u-blox 9 with protocol version 27	
Type	Output Message	
Comment	The GNSS receiver operating mode, satellites used for navigation, and DOP values. • If less than 12 SVs are used for navigation, the remaining fields are left empty. If more than 12 SVs are used for navigation, only the IDs of the first 12 are output. • The SV numbers (fields 'sv') are in the range of 1 to 32 for GPS satellites, and 33 to 64 for SBAS satellites (33 = SBAS PRN 120, 34 = SBAS PRN 121, and so on) **In a multi-GNSS system this message will be output multiple times, once for each GNSS.**	
	ID for CFG-MSG	*Number of fields*
Message Info	0xF0 0x02	21

Message Structure:

```
$xxGSA,opMode,navMode{,sv},PDOP,HDOP,VDOP,systemId*cs<CR><LF>
```

Example:

```
$GPGSA,A,3,23,29,07,08,09,18,26,28,,,,,1.94,1.18,1.54,1*0D
```

Field No.	*Name*	*Unit*	*Format*	*Example*	*Description*

GSA continued

Field No.	*Name*	*Unit*	*Format*	*Example*	*Description*
0	xxGSA	-	string	$GPGSA	GSA Message ID (xx = current Talker ID)
1	opMode	-	character	A	Operation mode, see first table below
2	navMode	-	digit	3	Navigation mode, see second table below and position fix flags description
Start of repeated block (12 times)					
3 + 1*N	sv	-	numeric	29	Satellite number
End of repeated block					
15	PDOP	-	numeric	1.94	Position dilution of precision
16	HDOP	-	numeric	1.18	Horizontal dilution of precision
17	VDOP	-	numeric	1.54	Vertical dilution of precision
18	systemId	-	numeric	1	NMEA defined GNSS System ID **NMEA v4.1 and above only**
19	cs	-	hexadecimal	*0D	Checksum
20	<CR><LF>	-	character	-	Carriage return and line feed

Table Operation Mode

Operation Mode	*Description*
M	Manually set to operate in 2D or 3D mode
A	Automatically switching between 2D or 3D mode

Table Navigation Mode

Navigation Mode	*Description, see also position fix flags description*
1	Fix not available
2	2D Fix
3	3D Fix

5.2.12 GST

5.2.12.1 GNSS Pseudo Range Error Statistics

Message	**GST**		
Description	**GNSS Pseudo Range Error Statistics**		
Firmware	Supported on: • u-blox 9 with protocol version 27		
Type	Output Message		
Comment	This message reports statistical information on the quality of the position solution.		
Message Info	*ID for CFG-MSG*	*Number of fields*	
	0xF0 0x07	11	

Message Structure:

```
$xxGST,time,rangeRms,stdMajor,stdMinor,orient,stdLat,stdLong,stdAlt*cs<CR><LF>
```

Example:

```
$GPGST,082356.00,1.8,,,,1.7,1.3,2.2*7E
```

Field No.	*Name*	*Unit*	*Format*	*Example*	*Description*

GST continued

Field No.	*Name*	*Unit*	*Format*	*Example*	*Description*
0	`xxGST`	-	string	$GPGST	GST Message ID (xx = current Talker ID)
1	`time`	-	hhmmss.ss	082356.00	UTC time of associated position fix, see note on UTC representation
2	`rangeRms`	m	numeric	1.8	RMS value of the standard deviation of the ranges
3	`stdMajor`	m	numeric	-	Standard deviation of semi-major axis (only supported in ADR 4.10 and above)
4	`stdMinor`	m	numeric	-	Standard deviation of semi-minor axis (only supported in ADR 4.10 and above)
5	`orient`	deg	numeric	-	Orientation of semi-major axis (only supported in ADR 4.10 and above)
6	`stdLat`	m	numeric	1.7	Standard deviation of latitude error
7	`stdLong`	m	numeric	1.3	Standard deviation of longitude error
8	`stdAlt`	m	numeric	2.2	Standard deviation of altitude error
9	`cs`	-	hexadecimal	*7E	Checksum
10	`<CR><LF>`	-	character	-	Carriage return and line feed

5.2.13 GSV

5.2.13.1 GNSS Satellites in View

Message	**GSV**	
Description	**GNSS Satellites in View**	
Firmware	Supported on: • u-blox 9 with protocol version 27	
Type	Output Message	
Comment	The number of satellites in view, together with each SV ID, elevation azimuth, and signal strength (C/No) value. Only four satellite details are transmitted in one message. **In a multi-GNSS system sets of GSV messages will be output multiple times, one set for each GNSS.**	
	ID for CFG-MSG	*Number of fields*
Message Info	0xF0 0x03	8..16

Message Structure:

```
$xxGSV,numMsg,msgNum,numSV,{,sv,elv,az,cno},signalId*cs<CR><LF>
```

Example:

```
$GPGSV,3,1,10,23,38,230,44,29,71,156,47,07,29,116,41,08,09,081,36,0*7F

$GPGSV,3,2,10,10,07,189,,05,05,220,,09,34,274,42,18,25,309,44,0*72

$GPGSV,3,3,10,26,82,187,47,28,43,056,46,0*77
```

Field No.	*Name*	*Unit*	*Format*	*Example*	*Description*
0	`xxGSV`	-	string	$GPGSV	GSV Message ID (xx = GSV Talker ID)
1	`numMsg`	-	digit	3	Number of messages, total number of GSV messages being output
2	`msgNum`	-	digit	1	Number of this message
3	`numSV`	-	numeric	10	Number of satellites in view

GSV continued

Field No.	*Name*	*Unit*	*Format*	*Example*	*Description*
Start of repeated block (1..4 times)					
4 + 4*N	`sv`	-	numeric	23	Satellite ID
5 + 4*N	`elv`	deg	numeric	38	Elevation (range 0-90)
6 + 4*N	`az`	deg	numeric	230	Azimuth, (range 0-359)
7 + 4*N	`cno`	dBHz	numeric	44	Signal strength (C/N0, range 0-99), blank when not tracking
End of repeated block					
5..16	`signalId`	-	numeric	0	NMEA defined GNSS Signal ID (0 = All signals, see Signal Identifiers table for other values) **NMEA v4.1 and above only**
6..16	`cs`	-	hexadecimal	*7F	Checksum
7..16	`<CR><LF>`	-	character	-	Carriage return and line feed

5.2.14 RMC

5.2.14.1 Recommended Minimum data

Message	**RMC**		
Description	**Recommended Minimum data**		
Firmware	Supported on: • u-blox 9 with protocol version 27		
Type	Output Message		
Comment	**The output of this message is dependent on the currently selected datum (default: WGS84)** The recommended minimum sentence defined by NMEA for GNSS system data.		
	ID for CFG-MSG	*Number of fields*	
Message Info	0xF0 0x04	16	

Message Structure:

```
$xxRMC,time,status,lat,NS,long,EW,spd,cog,date,mv,mvEW,posMode,navStatus*cs<CR><LF>
```

Example:

```
$GPRMC,083559.00,A,4717.11437,N,00833.91522,E,0.004,77.52,091202,,,A,V*57
```

Field No.	*Name*	*Unit*	*Format*	*Example*	*Description*
0	`xxRMC`	-	string	$GPRMC	RMC Message ID (xx = current Talker ID)
1	`time`	-	hhmmss.ss	083559.00	UTC time, see note on UTC representation
2	`status`	-	character	A	Status, V = Navigation receiver warning, A = Data valid, see position fix flags description
3	`lat`	-	ddmm.mmmmm	4717.11437	Latitude (degrees & minutes), see format description

RMC continued

Field No.	Name	Unit	Format	Example	Description
4	`NS`	-	character	N	North/South indicator
5	`long`	-	dddmm. mmmmm	00833.91522	Longitude (degrees & minutes), see format description
6	`EW`	-	character	E	East/West indicator
7	`spd`	knots	numeric	0.004	Speed over ground
8	`cog`	degrees	numeric	77.52	Course over ground
9	`date`	-	ddmmyy	091202	Date in day, month, year format, see note on UTC representation
10	`mv`	degrees	numeric	-	Magnetic variation value. Only supported in ADR 4.10 and above.
11	`mvEW`	-	character	-	Magnetic variation E/W indicator. Only supported in ADR 4.10 and above.
12	`posMode`	-	character	A	Mode Indicator, see position fix flags description **NMEA v2.3 and above only**
13	`navStatus`	-	character	V	Navigational status indicator (V = Equipment is not providing navigational status information) **NMEA v4.1 and above only**
14	`cs`	-	hexadecimal	*57	Checksum
15	`<CR><LF>`	-	character	-	Carriage return and line feed

5.2.15 TXT

5.2.15.1 Text Transmission

Message	**TXT**		
Description	**Text Transmission**		
Firmware	Supported on: • u-blox 9 with protocol version 27		
Type	Output Message		
Comment	**This message is not configured through UBX-CFG-MSG, but instead through `UBX-CFG-INF`.** This message outputs various information on the receiver, such as power-up screen, software version etc. This message can be configured using UBX Protocol message `UBX-CFG-INF`.		
Message Info	*ID for CFG-MSG*	*Number of fields*	
	0xF0 0x41	7	

Message Structure:

```
$xxTXT,numMsg,msgNum,msgType,text*cs<CR><LF>
```

Example:

```
$GPTXT,01,01,02,u-blox ag - www.u-blox.com*50

$GPTXT,01,01,02,ANTARIS ATR0620 HW 00000040*67
```

TXT continued

Field No.	*Name*	*Unit*	*Format*	*Example*	*Description*
Field No.	*Name*	*Unit*	*Format*	*Example*	*Description*
0	`xxTXT`	-	string	$GPTXT	TXT Message ID (xx = current Talker ID)
1	`numMsg`	-	numeric	01	Total number of messages in this transmission, 01..99
2	`msgNum`	-	numeric	01	Message number in this transmission, range 01..xx
3	`msgType`	-	numeric	02	Text identifier, u-blox receivers specify the type of the message with this number. 00: Error 01: Warning 02: Notice 07: User
4	`text`	-	string	www.u-blox.com	Any ASCII text
5	`cs`	-	hexadecimal	*67	Checksum
6	`<CR><LF>`	-	character	-	Carriage return and line feed

5.2.16 VLW

5.2.16.1 Dual ground/water distance

Message	**VLW**		
Description	**Dual ground/water distance**		
Firmware	Supported on: • u-blox 9 with protocol version 27		
Type	Output Message		
Comment	The distance traveled, relative to the water and over the ground. This message relates to the Odometer functionality.		
Message Info	*ID for CFG-MSG*	*Number of fields*	
	0xF0 0x0F	11	

Message Structure:

```
$xxVLW,twd,twdUnit,wd,wdUnit,tgd,tgdUnit,gd,gdUnit*cs<CR><LF>
```

Example:

```
$GPVLW,,N,,N,15.8,N,1.2,N*06
```

Field No.	*Name*	*Unit*	*Format*	*Example*	*Description*
0	`xxVLW`	-	string	$GPVLW	VLW Message ID (xx = current Talker ID)
1	`twd`	nm	numeric	-	Total cumulative water distance, not output
2	`twdUnit`	-	character	N	Fixed field: nautical miles
3	`wd`	nm	numeric	-	Water distance since reset, not output
4	`wdUnit`	-	character	N	Fixed field: nautical miles
5	`tgd`	nm	numeric	15.8	Total cumulative ground distance
6	`tgdUnit`	-	character	N	Fixed field: nautical miles
7	`gd`	nm	numeric	1.2	Ground distance since reset
8	`gdUnit`	-	character	N	Fixed field: nautical miles

VLW continued

Field No.	*Name*	*Unit*	*Format*	*Example*	*Description*
9	`cs`	-	hexadecimal	*06	Checksum
10	`<CR><LF>`	-	character	-	Carriage return and line feed

5.2.17 VTG

5.2.17.1 Course over ground and Ground speed

Message	**VTG**		
Description	**Course over ground and Ground speed**		
Firmware	Supported on: • u-blox 9 with protocol version 27		
Type	Output Message		
Comment	Velocity is given as Course over Ground (COG) and Speed over Ground (SOG).		
	ID for CFG-MSG	*Number of fields*	
Message Info	0xF0 0x05	12	

Message Structure:

```
$xxVTG,cogt,T,cogm,M,knots,N,kph,K,posMode*cs<CR><LF>
```

Example:

```
$GPVTG,77.52,T,,M,0.004,N,0.008,K,A*06
```

Field No.	*Name*	*Unit*	*Format*	*Example*	*Description*
0	`xxVTG`	-	string	$GPVTG	VTG Message ID (xx = current Talker ID)
1	`cogt`	degrees	numeric	77.52	Course over ground (true)
2	`T`	-	character	T	Fixed field: true
3	`cogm`	degrees	numeric	-	Course over ground (magnetic). Only supported in ADR 4.10 and above.
4	`M`	-	character	M	Fixed field: magnetic
5	`knots`	knots	numeric	0.004	Speed over ground
6	`N`	-	character	N	Fixed field: knots
7	`kph`	km/h	numeric	0.008	Speed over ground
8	`K`	-	character	K	Fixed field: kilometers per hour
9	`posMode`	-	character	A	Mode Indicator, see position fix flags description **NMEA v2.3 and above only**
10	`cs`	-	hexadecimal	*06	Checksum
11	`<CR><LF>`	-	character	-	Carriage return and line feed

5.2.18 ZDA

5.2.18.1 Time and Date

Message	**ZDA**		
Description	**Time and Date**		
Firmware	Supported on: • u-blox 9 with protocol version 27		
Type	Output Message		
Comment	-		
Message Info	*ID for CFG-MSG*	*Number of fields*	
	0xF0 0x08	9	

Message Structure:

```
$xxZDA,hhmmss.ss,day,month,year,ltzh,ltzn*cs<CR><LF>
```

Example:

```
$GPZDA,082710.00,16,09,2002,00,00*64
```

Field No.	*Name*	*Unit*	*Format*	*Example*	*Description*
0	`xxZDA`	-	string	$GPZDA	ZDA Message ID (xx = current Talker ID)
1	`time`	-	hhmmss.ss	082710.00	UTC Time, see note on UTC representation
2	`day`	day	dd	16	UTC day (range: 1-31)
3	`month`	month	mm	09	UTC month (range: 1-12)
4	`year`	year	yyyy	2002	UTC year
5	`ltzh`	-	-xx	00	Local time zone hours (fixed to 00)
6	`ltzn`	-	zz	00	Local time zone minutes (fixed to 00)
7	`cs`	-	hexadecimal	*64	Checksum
8	`<CR><LF>`	-	character	-	Carriage return and line feed

6 UBX Protocol

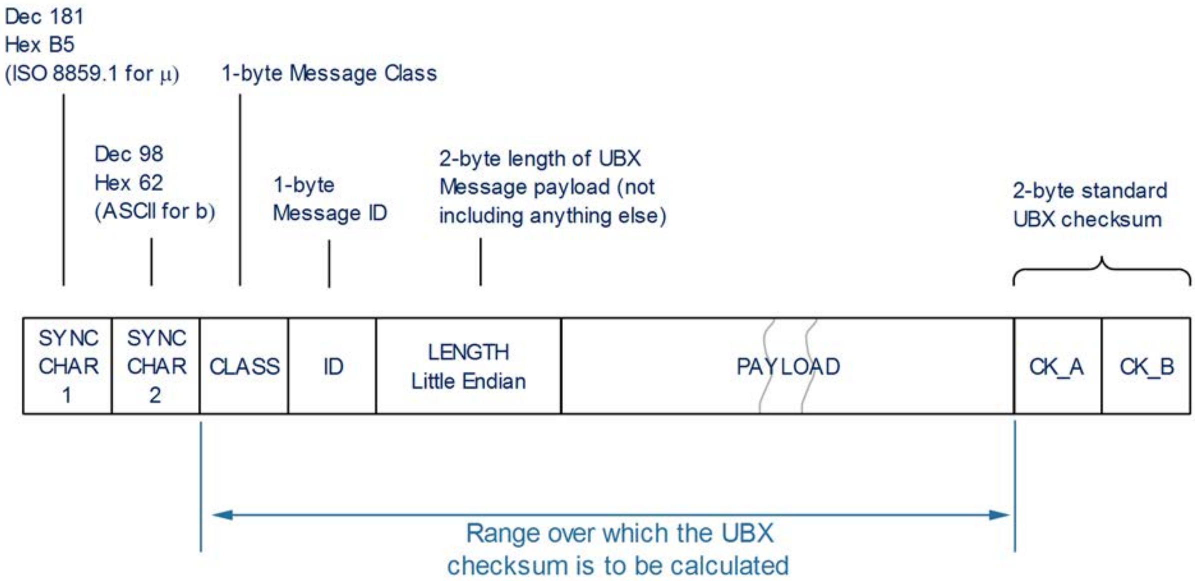
6.1 UBX Protocol Key Features

u-blox receivers support a u-blox proprietary protocol to communicate with a host computer. This protocol has the following key features:

- Compact - uses 8-bit Binary Data.
- Checksum Protected - uses a low-overhead checksum algorithm
- Modular - uses a 2-stage message identifier (Class and Message ID)

6.2 UBX Frame Structure

The structure of a basic UBX Frame is shown in the following diagram.



- Every **Frame** starts with a 2-byte Preamble consisting of two synchronization characters: 0xB5 0x62.
- A 1-byte *Message* **Class** field follows. A Class is a group of messages that are related to each other.
- A 1-byte *Message* **ID** field defines the message that is to follow.
- A 2-byte **Length** field follows. The length is defined as being that of the payload only. It does not include the Preamble, Message Class, Message ID, Length, or CRC fields. The number format of the length field is a Little-Endian unsigned 16-bit integer.
- The **Payload** field contains a variable number of bytes.
- The two 1-byte **CK_A** and **CK_B** fields hold a 16-bit checksum whose calculation is defined below. This concludes the Frame.

6.3 UBX Payload Definition Rules

6.3.1 Structure Packing

Values are placed in an order that structure packing is not a problem. This means that 2-byte values shall start on offsets which are a multiple of 2; 4-byte values shall start at a multiple of 4; and so on.

6.3.2 Reserved Elements

Some messages contain reserved fields or bits to allow for future expansion. The contents of these elements should be ignored in output messages and must be set to zero in input messages. Where a message is output and subsequently returned to the receiver as input message, reserved elements can either be explicitly set to zero or left with whatever value they were output with.

6.3.3 Undefined Values

The description of some fields provide specific meanings for specific values. For example, the field gnssId appears in many UBX messages and uses 0 to indicate GPS, 1 for SBAS and so on (see Satellite Numbering for details); however it is usually stored in a byte with far more possible values than the handful currently defined. All such undefined values are reserved for future expansion and therefore should not be used.

6.3.4 Message Naming

Referring to messages is done by adding the class name and a dash in front of the message name. For example, the version information message is referred to as `UBX-MON-VER`. Referring to message fields or their values is done by adding a dot and the name, e.g. `UBX-MON-VER.swVersion`.

6.3.5 Number Formats

All multi-byte values are ordered in Little Endian format, unless otherwise indicated.

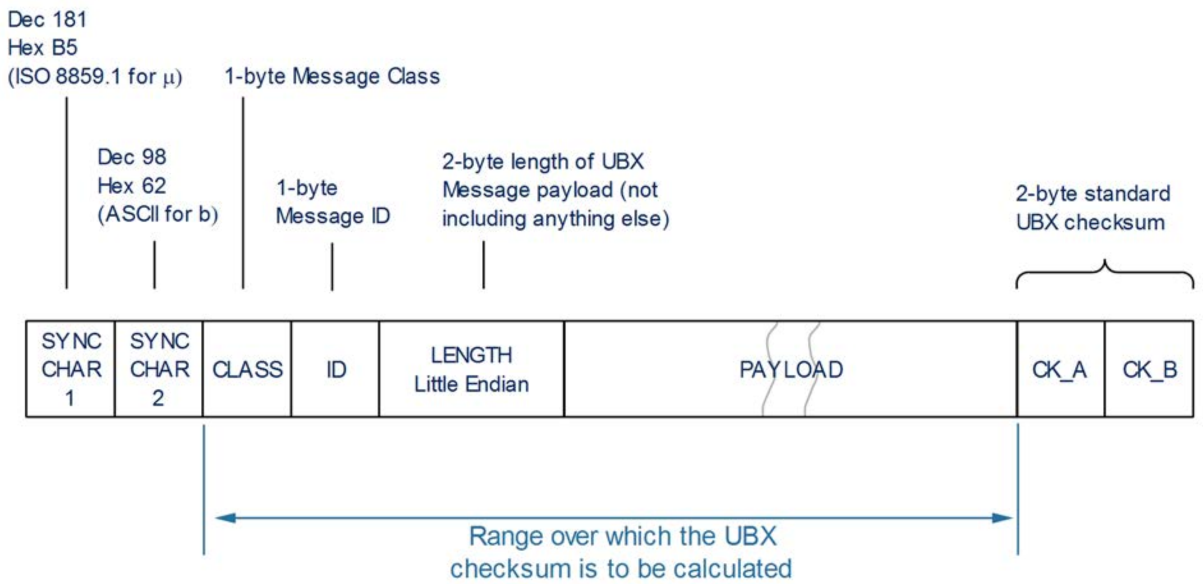
All floating point values are transmitted in IEEE754 single or double precision.

Variable Type Definitions

Short	*Type*	*Size (Bytes)*	*Comment*	*Min/Max*	*Resolution*
U1	Unsigned Char	1		0..255	1
RU1_3	Unsigned Char	1	binary floating point with 3 bit exponent, eeeb bbbb, (Value & 0x1F) << (Value >> 5)	0..(31*2^7) non-continuous	~ 2^(Value >> 5)
I1	Signed Char	1	2's complement	-128 .. 127	1
X1	Bitfield	1		n/a	n/a
U2	Unsigned Short	2		0 .. 65535	1
I2	Signed Short	2	2's complement	-32768 .. 32767	1
X2	Bitfield	2		n/a	n/a
U4	Unsigned Long	4		0 .. 4'294'967'295	1
I4	Signed Long	4	2's complement	-2'147'483'648 .. 2'147'483'647	1
X4	Bitfield	4		n/a	n/a
R4	IEEE 754 Single Precision	4		-1*2^+127 .. 2^+127	~ Value * 2^-24
R8	IEEE 754 Double Precision	8		-1*2^+1023 .. 2^+1023	~ Value * 2^-53
CH	ASCII / ISO 8859.1 Encoding	1			

6.4 UBX Checksum

The checksum is calculated over the Message, starting and including the CLASS field, up until, but excluding, the Checksum Field:



The checksum algorithm used is the 8-Bit Fletcher Algorithm, which is used in the TCP standard (RFC 1145). This algorithm works as follows:

Buffer[N] contains the data over which the checksum is to be calculated.

The two CK_ values are 8-Bit unsigned integers, only! If implementing with larger-sized integer values, make sure to mask both CK_A and CK_B with 0xFF after both operations in the loop.

```
CK_A = 0, CK_B = 0
For(I=0;I<N;I++)
{
    CK_A = CK_A + Buffer[I]
    CK_B = CK_B + CK_A
}
```

After the loop, the two U1 values contain the checksum, transmitted after the Message, which conclude the Frame.

6.5 UBX Message Flow

There are certain features associated with the messages being sent back and forth:

6.5.1 Acknowledgement

When messages from the class CFG are sent to the receiver, the receiver will send an "acknowledge" (UBX-ACK-ACK) or a "not acknowledge" (UBX-ACK-NAK) message back to the sender, depending on whether or not the message was processed correctly.

Some messages from other classes (e.g. LOG) also use the same acknowledgement mechanism.

6.5.2 Polling Mechanism

All messages that are output by the receiver in a periodic manner (i.e. messages in classes MON, NAV and RXM) and Get/Set type messages, such as the configuration messages in the CFG class, can also be polled.

The UBX protocol is designed so that messages can be polled by sending the message required to the receiver but without a payload (or with just a single parameter that identifies the poll request). The receiver then responds with the same message with the payload populated.

6.6 UBX Class IDs

A class is a grouping of messages which are related to each other. The following table lists all the current message classes.

Name	*Class*	*Description*
NAV	0x01	Navigation Results Messages: Position, Speed, Time, Acceleration, Heading, DOP, SVs used
RXM	0x02	Receiver Manager Messages: Satellite Status, RTC Status
INF	0x04	Information Messages: Printf-Style Messages, with IDs such as Error, Warning, Notice
ACK	0x05	Ack/Nak Messages: Acknowledge or Reject messages to UBX-CFG input messages
CFG	0x06	Configuration Input Messages: Set Dynamic Model, Set DOP Mask, Set Baud Rate, etc.
UPD	0x09	Firmware Update Messages: Memory/Flash erase/write, Reboot, Flash identification, etc.
MON	0x0A	Monitoring Messages: Communication Status, CPU Load, Stack Usage, Task Status
TIM	0x0D	Timing Messages: Time Pulse Output, Time Mark Results
MGA	0x13	Multiple GNSS Assistance Messages: Assistance data for various GNSS
LOG	0x21	Logging Messages: Log creation, deletion, info and retrieval
SEC	0x27	Security Feature Messages

All remaining class IDs are reserved.

6.7 UBX Messages Overview

Page	Mnemonic	Cls/ID	Length	Type	Description
	UBX Class ACK			**Ack/Nak Messages**	
47	**ACK-ACK**	0x05 0x01	2	Output	Message Acknowledged
47	**ACK-NAK**	0x05 0x00	2	Output	Message Not-Acknowledged
	UBX Class CFG			**Configuration Input Messages**	
48	**CFG-ANT**	0x06 0x13	4	Get/Set	Antenna Control Settings
49	**CFG-CFG**	0x06 0x09	(12) or (13)	Command	Clear, Save and Load configurations
51	**CFG-DAT**	0x06 0x06	44	Set	Set User-defined Datum.
52	**CFG-DAT**	0x06 0x06	52	Get	The currently defined Datum
53	**CFG-DGNSS**	0x06 0x70	4	Get/Set	DGNSS configuration
54	**CFG-DYNSEED**	0x06 0x85	12	Set	Programming the dynamic seed for the host...
54	**CFG-FIXSEED**	0x06 0x84	12 + 2*length	Set	Programming the fixed seed for host...
55	**CFG-GEOFENCE**	0x06 0x69	8 + 12*numFe...	Get/Set	Geofencing configuration
56	**CFG-GNSS**	0x06 0x3E	4 + 8*numCo...	Get/Set	GNSS system configuration
58	**CFG-INF**	0x06 0x02	1	Poll Request	Poll configuration for one protocol
59	**CFG-INF**	0x06 0x02	0 + 10*N	Get/Set	Information message configuration
60	**CFG-ITFM**	0x06 0x39	8	Get/Set	Jamming/Interference Monitor configuration
61	**CFG-LOGFILTER**	0x06 0x47	12	Get/Set	Data Logger Configuration
63	**CFG-MSG**	0x06 0x01	2	Poll Request	Poll a message configuration
63	**CFG-MSG**	0x06 0x01	8	Get/Set	Set Message Rate(s)
64	**CFG-MSG**	0x06 0x01	3	Get/Set	Set Message Rate
64	**CFG-NAV5**	0x06 0x24	36	Get/Set	Navigation Engine Settings
66	**CFG-NAVX5**	0x06 0x23	40	Get/Set	Navigation Engine Expert Settings
69	**CFG-NMEA**	0x06 0x17	20	Get/Set	Extended NMEA protocol configuration V1
72	**CFG-ODO**	0x06 0x1E	20	Get/Set	Odometer, Low-speed COG Engine Settings
73	**CFG-OTP**	0x06 0x41	0	Poll Request	Poll OTP content
73	**CFG-OTP**	0x06 0x41	128	Get	OTP content
74	**CFG-OTP**	0x06 0x41	0 + 1*N	Set	Writes OTP content
74	**CFG-OTP**	0x06 0x41	10	Set	Write the USB vendor ID file (0x20)
75	**CFG-OTP**	0x06 0x41	8 + 1*length	Set	Write the USB vendor string file (0x21)
76	**CFG-OTP**	0x06 0x41	10	Set	Write the USB product ID file (0x22)
76	**CFG-OTP**	0x06 0x41	8 + 1*length	Set	Write the USB product string file (0x23)
77	**CFG-OTP**	0x06 0x41	9	Set	Write the receiver configuration file (0x30)
78	**CFG-OTP**	0x06 0x41	9	Set	Write the post production flags file (0x32)
79	**CFG-OTP**	0x06 0x41	13	Set	Write the oscillator offset calibration file (0x36)
80	**CFG-PIO**	0x06 0x2c	19	Set	Set PIO pins
81	**CFG-PRT**	0x06 0x00	1	Poll Request	Polls the configuration for one I/O Port
82	**CFG-PRT**	0x06 0x00	20	Get/Set	Port Configuration for UART

UBX Messages Overview continued

Page	*Mnemonic*	*Cls/ID*	*Length*	*Type*	*Description*
85	**CFG-PRT**	0x06 0x00	20	Get/Set	Port Configuration for USB Port
87	**CFG-PRT**	0x06 0x00	20	Get/Set	Port Configuration for SPI Port
89	**CFG-PRT**	0x06 0x00	20	Get/Set	Port Configuration for DDC Port
92	**CFG-PT2**	0x06 0x59	12 + 4*N	Set	Production test configuration
93	**CFG-PWR**	0x06 0x57	8	Set	Put receiver in a defined power state.
94	**CFG-RATE**	0x06 0x08	6	Get/Set	Navigation/Measurement Rate Settings
95	**CFG-RINV**	0x06 0x34	1 + 1*N	Get/Set	Contents of Remote Inventory
96	**CFG-RST**	0x06 0x04	4	Command	Reset Receiver / Clear Backup Data Structures
97	**CFG-TMODE3**	0x06 0x71	40	Get/Set	Time Mode Settings 3
99	**CFG-TP5**	0x06 0x31	32	Get/Set	Time Pulse Parameters
101	**CFG-USBTEST**	0x06 0x58	2	Set	USB Testing
101	**CFG-USB**	0x06 0x1B	108	Get/Set	USB Configuration
102	**CFG-VALDEL**	0x06 0x8C	4 + 4*N	Set	Deletes values corresponding to provided keys
103	**CFG-VALDEL**	0x06 0x8C	4 + 4*N	Set	Deletes values corresponding to provided...
105	**CFG-VALGET**	0x06 0x8B	4 + 4*N	Poll Request	Get Configuration Items
106	**CFG-VALGET**	0x06 0x8B	4 + 1*N	polled	Configuration Items
107	**CFG-VALSET**	0x06 0x8A	4 + 1*N	Set	Sets values corresponding to provided...
108	**CFG-VALSET**	0x06 0x8A	4 + 1*N	Set	Sets values corresponding to provided...
UBX Class INF				**Information Messages**	
110	**INF-DEBUG**	0x04 0x04	0 + 1*N	Output	ASCII output with debug contents
110	**INF-ERROR**	0x04 0x00	0 + 1*N	Output	ASCII output with error contents
111	**INF-NOTICE**	0x04 0x02	0 + 1*N	Output	ASCII output with informational contents
111	**INF-TEST**	0x04 0x03	0 + 1*N	Output	ASCII output with test contents
112	**INF-WARNING**	0x04 0x01	0 + 1*N	Output	ASCII output with warning contents
UBX Class LOG				**Logging Messages**	
113	**LOG-CREATE**	0x21 0x07	8	Command	Create Log File
114	**LOG-ERASE**	0x21 0x03	0	Command	Erase Logged Data
114	**LOG-FINDTIME**	0x21 0x0E	12	Input	Find index of a log entry based on a given time
115	**LOG-FINDTIME**	0x21 0x0E	8	Output	Response to FINDTIME request.
115	**LOG-INFO**	0x21 0x08	0	Poll Request	Poll for log information
116	**LOG-INFO**	0x21 0x08	48	Output	Log information
117	**LOG-RETRIEVEPOSE...**	0x21 0x0f	32	Output	Odometer log entry
118	**LOG-RETRIEVEPOS**	0x21 0x0b	40	Output	Position fix log entry
119	**LOG-RETRIEVESTRING**	0x21 0x0d	16 + 1*byteC...	Output	Byte string log entry
119	**LOG-RETRIEVE**	0x21 0x09	12	Command	Request log data
120	**LOG-STRING**	0x21 0x04	0 + 1*N	Command	Store arbitrary string in on-board flash
UBX Class MGA				**Multiple GNSS Assistance Messages**	
121	**MGA-ACK-DATA0**	0x13 0x60	8	Output	Multiple GNSS Acknowledge message

UBX Messages Overview continued

Page	*Mnemonic*	*Cls/ID*	*Length*	*Type*	*Description*
122	**MGA-BDS-EPH**	0x13 0x03	88	Input	BDS Ephemeris Assistance
123	**MGA-BDS-ALM**	0x13 0x03	40	Input	BDS Almanac Assistance
124	**MGA-BDS-HEALTH**	0x13 0x03	68	Input	BDS Health Assistance
124	**MGA-BDS-UTC**	0x13 0x03	20	Input	BDS UTC Assistance
125	**MGA-BDS-IONO**	0x13 0x03	16	Input	BDS Ionospheric Assistance
126	**MGA-DBD**	0x13 0x80	0	Poll Request	Poll the Navigation Database
126	**MGA-DBD**	0x13 0x80	12 + 1*N	Input/Output	Navigation Database Dump Entry
127	**MGA-GAL-EPH**	0x13 0x02	76	Input	Galileo Ephemeris Assistance
128	**MGA-GAL-ALM**	0x13 0x02	32	Input	Galileo Almanac Assistance
129	**MGA-GAL-TIMEOFF...**	0x13 0x02	12	Input	Galileo GPS time offset assistance
130	**MGA-GAL-UTC**	0x13 0x02	20	Input	Galileo UTC Assistance
130	**MGA-GLO-EPH**	0x13 0x06	48	Input	GLONASS Ephemeris Assistance
132	**MGA-GLO-ALM**	0x13 0x06	36	Input	GLONASS Almanac Assistance
133	**MGA-GLO-TIMEOFF...**	0x13 0x06	20	Input	GLONASS Auxiliary Time Offset Assistance
133	**MGA-GPS-EPH**	0x13 0x00	68	Input	GPS Ephemeris Assistance
135	**MGA-GPS-ALM**	0x13 0x00	36	Input	GPS Almanac Assistance
136	**MGA-GPS-HEALTH**	0x13 0x00	40	Input	GPS Health Assistance
136	**MGA-GPS-UTC**	0x13 0x00	20	Input	GPS UTC Assistance
137	**MGA-GPS-IONO**	0x13 0x00	16	Input	GPS Ionosphere Assistance
138	**MGA-INI-POS_XYZ**	0x13 0x40	20	Input	Initial Position Assistance
138	**MGA-INI-POS_LLH**	0x13 0x40	20	Input	Initial Position Assistance
139	**MGA-INI-TIME_UTC**	0x13 0x40	24	Input	Initial Time Assistance
140	**MGA-INI-TIME_GNSS**	0x13 0x40	24	Input	Initial Time Assistance
141	**MGA-INI-CLKD**	0x13 0x40	12	Input	Initial Clock Drift Assistance
142	**MGA-INI-FREQ**	0x13 0x40	12	Input	Initial Frequency Assistance
143	**MGA-INI-EOP**	0x13 0x40	72	Input	Earth Orientation Parameters Assistance
143	**MGA-QZSS-EPH**	0x13 0x05	68	Input	QZSS Ephemeris Assistance
145	**MGA-QZSS-ALM**	0x13 0x05	36	Input	QZSS Almanac Assistance
146	**MGA-QZSS-HEALTH**	0x13 0x05	12	Input	QZSS Health Assistance
	UBX Class MON			**Monitoring Messages**	
147	**MON-COMMS**	0x0A 0x36	8 + 40*nPorts	Periodic/Polled	Comm port information
148	**MON-GNSS**	0x0A 0x28	8	Polled	Information message major GNSS selection
150	**MON-HW2**	0x0A 0x0B	28	Periodic/Polled	Extended Hardware Status
151	**MON-HW3**	0x0A 0x37	22 + 6*nPins	Periodic/Polled	HW I/O pin information
152	**MON-HW**	0x0A 0x09	60	Periodic/Polled	Hardware Status
154	**MON-IO**	0x0A 0x02	0 + 20*N	Periodic/Polled	I/O Subsystem Status
154	**MON-MSGPP**	0x0A 0x06	120	Periodic/Polled	Message Parse and Process Status
155	**MON-PATCH**	0x0A 0x27	0	Poll Request	Poll Request for installed patches

UBX Messages Overview continued

Page	*Mnemonic*	*Cls/ID*	*Length*	*Type*	*Description*
155	**MON-PATCH**	0x0A 0x27	4 + 16*nEntries	Polled	Output information about installed patches.
156	**MON-PIO**	0x0A 0x24	19 + 17	Polled	Production testing message for PIO pins
158	**MON-PT2**	0x0A 0x2B	24 + 28*num...	Periodic/Polled	Multi-GNSS Production Test Monitor Message
160	**MON-RF**	0x0A 0x38	4 + 24*nBlocks	Periodic/Polled	RF information
161	**MON-RXBUF**	0x0A 0x07	24	Periodic/Polled	Receiver Buffer Status
161	**MON-RXR**	0x0A 0x21	1	Output	Receiver Status Information
162	**MON-TEMP**	0x0A 0x0E	0	Poll Request	Poll Temperature value [C] and temperature...
162	**MON-TEMP**	0x0A 0x0E	12	Periodic/Polled	Temperature value [C] and temperature...
163	**MON-TXBUF**	0x0A 0x08	28	Periodic/Polled	Transmitter Buffer Status
164	**MON-VER**	0x0A 0x04	0	Poll Request	Poll Receiver/Software Version
164	**MON-VER**	0x0A 0x04	40 + 30*N	Polled	Receiver/Software Version
	UBX Class NAV			**Navigation Results Messages**	
165	**NAV-CLOCK**	0x01 0x22	20	Periodic/Polled	Clock Solution
165	**NAV-COV**	0x01 0x36	64	Periodic/Polled	Covariance matrices
166	**NAV-DOP**	0x01 0x04	18	Periodic/Polled	Dilution of precision
167	**NAV-EOE**	0x01 0x61	4	Periodic	End Of Epoch
167	**NAV-GEOFENCE**	0x01 0x39	8 + 2*numFen...	Periodic/Polled	Geofencing status
168	**NAV-HPPOSECEF**	0x01 0x13	28	Periodic/Polled	High Precision Position Solution in ECEF
169	**NAV-HPPOSLLH**	0x01 0x14	36	Periodic/Polled	High Precision Geodetic Position Solution
170	**NAV-ODO**	0x01 0x09	20	Periodic/Polled	Odometer Solution
170	**NAV-ORB**	0x01 0x34	8 + 6*numSv	Periodic/Polled	GNSS Orbit Database Info
173	**NAV-POSECEF**	0x01 0x01	20	Periodic/Polled	Position Solution in ECEF
174	**NAV-POSLLH**	0x01 0x02	28	Periodic/Polled	Geodetic Position Solution
174	**NAV-PVT**	0x01 0x07	92	Periodic/Polled	Navigation Position Velocity Time Solution
177	**NAV-RELPOSNED**	0x01 0x3C	40	Periodic/Polled	Relative Positioning Information in NED frame
179	**NAV-RESETODO**	0x01 0x10	0	Command	Reset odometer
179	**NAV-SAT**	0x01 0x35	8 + 12*numSvs	Periodic/Polled	Satellite Information
181	**NAV-SIG**	0x01 0x43	8 + 16*numSigs	Periodic/Polled	Signal Information
183	**NAV-STATUS**	0x01 0x03	16	Periodic/Polled	Receiver Navigation Status
185	**NAV-SVIN**	0x01 0x3B	40	Periodic/Polled	Survey-in data
186	**NAV-TIMEBDS**	0x01 0x24	20	Periodic/Polled	BDS Time Solution
187	**NAV-TIMEGAL**	0x01 0x25	20	Periodic/Polled	Galileo Time Solution
188	**NAV-TIMEGLO**	0x01 0x23	20	Periodic/Polled	GLO Time Solution
189	**NAV-TIMEGPS**	0x01 0x20	16	Periodic/Polled	GPS Time Solution
190	**NAV-TIMELS**	0x01 0x26	24	Periodic/Polled	Leap second event information
192	**NAV-TIMEUTC**	0x01 0x21	20	Periodic/Polled	UTC Time Solution
193	**NAV-VELECEF**	0x01 0x11	20	Periodic/Polled	Velocity Solution in ECEF
194	**NAV-VELNED**	0x01 0x12	36	Periodic/Polled	Velocity Solution in NED

UBX Messages Overview continued

Page	*Mnemonic*	*Cls/ID*	*Length*	*Type*	*Description*
UBX Class RXM				**Receiver Manager Messages**	
195	**RXM-MEASX**	0x02 0x14	44 + 24*numSV	Periodic	Satellite Measurements for RRLP
197	**RXM-PMREQ**	0x02 0x41	8	Command	Requests a Power Management task
197	**RXM-PMREQ**	0x02 0x41	16	Command	Requests a Power Management task
199	**RXM-RAWX**	0x02 0x15	16 + 32*num...	Periodic/Polled	Multi-GNSS Raw Measurement Data
202	**RXM-RLM**	0x02 0x59	16	Output	Galileo SAR Short-RLM report
203	**RXM-RLM**	0x02 0x59	28	Output	Galileo SAR Long-RLM report
203	**RXM-RTC5**	0x02 0x23	28	Periodic/Polled	Real Time Clock Status
204	**RXM-RTCM**	0x02 0x32	8	Output	RTCM input status
205	**RXM-SFRBX**	0x02 0x13	8 + 4*numWo...	Output	Broadcast Navigation Data Subframe
UBX Class SEC				**Security Feature Messages**	
206	**SEC-SIGN**	0x27 0x01	40	Output	Signature of a previous message
206	**SEC-UNIQID**	0x27 0x03	9	Output	Unique Chip ID
UBX Class TIM				**Timing Messages**	
207	**TIM-TM2**	0x0D 0x03	28	Periodic/Polled	Time mark data
208	**TIM-TP**	0x0D 0x01	16	Periodic/Polled	Time Pulse Timedata
210	**TIM-VRFY**	0x0D 0x06	20	Periodic/Polled	Sourced Time Verification
UBX Class UPD				**Firmware Update Messages**	
211	**UPD-CERASE**	0x09 0x16	0	Command	Chip erase the connected SQI flash
211	**UPD-CERASE**	0x09 0x16	1	Output	Chip erase the connected SQI flash
212	**UPD-CRC**	0x09 0x0D	16	Command	Check CRC over firmware in flash.
212	**UPD-CRC**	0x09 0x0D	5	Output	Result of CRC check.
213	**UPD-ERASE**	0x09 0x0B	4	Command	Erase flash sector
213	**UPD-ERASE**	0x09 0x0B	5	Output	Erase flash sector
214	**UPD-FLDET**	0x09 0x08	4	Poll Request	Get the Flash manufacturer and device IDs
214	**UPD-FLDET**	0x09 0x08	8	Get	Get the Flash manufacturer and device IDs
215	**UPD-FLWRI**	0x09 0x0C	8 + 1*size	Command	Write flash data (area must be erased before)
215	**UPD-FLWRI**	0x09 0x0C	5	Output	Write flash data success indication
216	**UPD-IDEN**	0x09 0x06	0	Poll Request	Identify flash loader version
216	**UPD-IDEN**	0x09 0x06	1	Get	Identify flash loader version
216	**UPD-POS**	0x09 0x15	2	Command	Enable PLL during safeboot
217	**UPD-QSIZE**	0x09 0x09	0	Poll Request	Get number of pending commands in queue
217	**UPD-QSIZE**	0x09 0x09	1	Get	Number of pending commands in queue
218	**UPD-QSIZE**	0x09 0x09	2	Get	Number of pending commands in queue
218	**UPD-RBOOT**	0x09 0x0E	0	Command	Performs a watchdog reset
218	**UPD-ROM**	0x09 0x25	12	Polled	Message is holding ROM CRC
219	**UPD-SAFE**	0x09 0x07	0	Command	Boot in safe environment from ROM or RAM
219	**UPD-SAFE**	0x09 0x07	1	Command	Start flash loader task

UBX Messages Overview continued

Page	*Mnemonic*	*Cls/ID*	*Length*	*Type*	*Description*
220	**UPD-SETQ**	0x09 0x0F	1	Set	Set maximum of pending commands in queue
220	**UPD-SETQ**	0x09 0x0F	2	Set	Set maximum of pending commands in queue
221	**UPD-SOS**	0x09 0x14	0	Poll Request	Poll Backup File Restore Status
221	**UPD-SOS**	0x09 0x14	4	Command	Create Backup File in Flash
222	**UPD-SOS**	0x09 0x14	4	Command	Clear Backup in Flash
222	**UPD-SOS**	0x09 0x14	8	Output	Backup File Creation Acknowledge
223	**UPD-SOS**	0x09 0x14	8	Output	System Restored from Backup

6.8 UBX-ACK (0x05)

Ack/Nak Messages: i.e. Acknowledge or Reject messages to UBX-CFG input messages.
Messages in the UBX-ACK class output the processing results to UBX-CFG and some other messages.

6.8.1 UBX-ACK-ACK (0x05 0x01)

6.8.1.1 Message Acknowledged

Message	**UBX-ACK-ACK**
Description	**Message Acknowledged**
Firmware	Supported on: • u-blox 9 with protocol version 27
Type	Output
Comment	Output upon processing of an input message. ACK Message is sent as soon as possible but at least within one second.

	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x05	0x01	2	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`clsID`	-	Class ID of the Acknowledged Message
1	U1	-	`msgID`	-	Message ID of the Acknowledged Message

6.8.2 UBX-ACK-NAK (0x05 0x00)

6.8.2.1 Message Not-Acknowledged

Message	**UBX-ACK-NAK**
Description	**Message Not-Acknowledged**
Firmware	Supported on: • u-blox 9 with protocol version 27
Type	Output
Comment	Output upon processing of an input message. NAK Message is sent as soon as possible but at least within one second.

	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x05	0x00	2	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`clsID`	-	Class ID of the Not-Acknowledged Message
1	U1	-	`msgID`	-	Message ID of the Not-Acknowledged Message

6.9 UBX-CFG (0x06)

Configuration Input Messages: i.e. Set Dynamic Model, Set DOP Mask, Set Baud Rate, etc..
Messages in the CFG class are used to configure the receiver and read out current configuration values. Any messages in the CFG class sent to the receiver are either acknowledged (with message UBX-ACK-ACK) if processed successfully or rejected (with message UBX-ACK-NAK) if processing unsuccessfully.

6.9.1 UBX-CFG-ANT (0x06 0x13)

6.9.1.1 Antenna Control Settings

Message	**UBX-CFG-ANT**
Description	**Antenna Control Settings**
Firmware	Supported on: • u-blox 9 with protocol version 27
Type	Get/Set
Comment	**This message is deprecated in protocol versions greater than 23.01. Use UBX-CFG-VALSET, UBX-CFG-VALGET, UBX-CFG-VALDEL instead.** See the Legacy UBX Message Fields Reference for the corresponding configuration item. This message allows the user to configure the antenna supervisor. The antenna supervisor can be used to detect the status of an active antenna and control it. It can be used to turn off the supply to the antenna in the event of a short (for example) or to manage power consumption in Power Save Mode. Refer to Antenna Supervisor Configuration and the relevant Hardware Integration Manual (HIM) for more information regarding the behavior of the antenna supervisor. Refer to UBX-MON-HW for a description of the fields in the message used to obtain the status of the antenna. Note that not all pins can be used for antenna supervisor operation, it is recommended that you use the default pins, consult the Integration Manual if you need to use other pins.

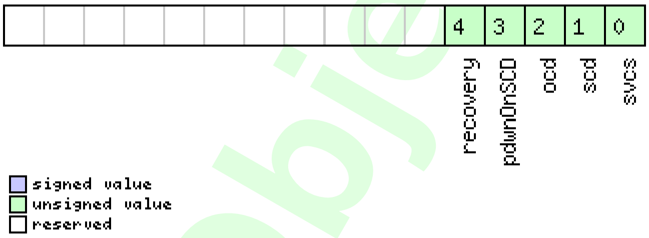
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x06	0x13	4	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	X2	-	`flags`	-	Antenna Flag Mask (see graphic below)
2	X2	-	`pins`	-	Antenna Pin Configuration (see graphic below)

Bitfield flags

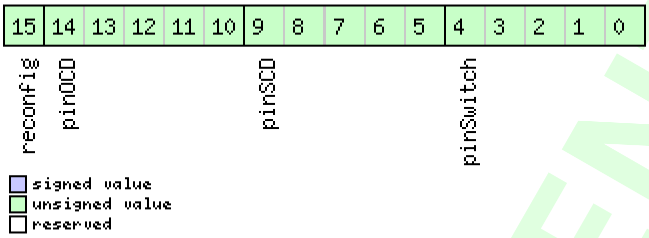
This graphic explains the bits of `flags`



Name	*Description*
`svcs`	Enable Antenna Supply Voltage Control Signal
`scd`	Enable Short Circuit Detection
`ocd`	Enable Open Circuit Detection
`pdwnOnSCD`	Power Down Antenna supply if Short Circuit is detected. (only in combination with Bit 1)
`recovery`	Enable automatic recovery from short state

Bitfield pins

This graphic explains the bits of `pins`



Name	*Description*
`pinSwitch`	PIO-Pin used for switching antenna supply
`pinSCD`	PIO-Pin used for detecting a short in the antenna supply
`pinOCD`	PIO-Pin used for detecting open/not connected antenna
`reconfig`	if set to one, and this command is sent to the receiver, the receiver will reconfigure the pins as specified.

6.9.2 UBX-CFG-CFG (0x06 0x09)

6.9.2.1 Clear, Save and Load configurations

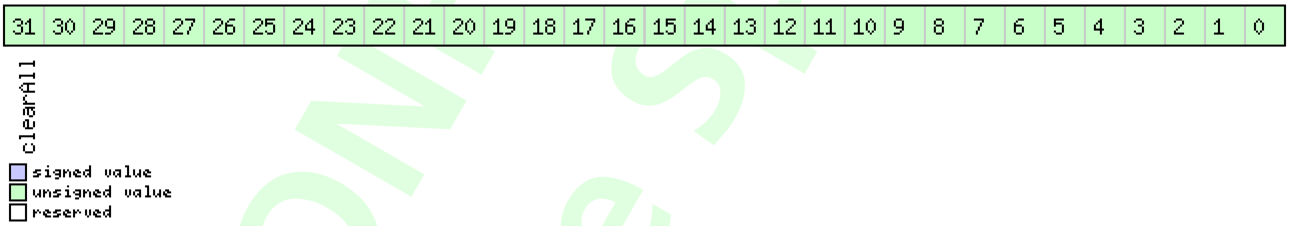
<table>
<tr><td><i>Message</i></td><td colspan="6"><b>UBX-CFG-CFG</b></td></tr>
<tr><td><i>Description</i></td><td colspan="6"><b>Clear, Save and Load configurations</b></td></tr>
<tr><td><i>Firmware</i></td><td colspan="6">Supported on:
• u-blox 9 with protocol version 27</td></tr>
<tr><td><i>Type</i></td><td colspan="6">Command</td></tr>
<tr><td><i>Comment</i></td><td colspan="6"><b>This message is deprecated in protocol versions greater than 23.01. Use <code>UBX-CFG-VALSET</code>, <code>UBX-CFG-VALGET</code>, <code>UBX-CFG-VALDEL</code> instead.</b>
See Receiver Configuration for a detailed description on how Receiver Configuration should be used. The behaviour of this message has changed. The three masks which were used to clear, save and load a subsection of configuration lost their meaning. It is no longer possible to save or clear a subsection of the configuration using this message. The behaviour of the masks is described as follows :
• if any bit is set in the clearMask: all configuration in the selected non-volatile memory is deleted
• if any bit is set in the saveMask: all current configuration is stored (copied) to the selected layers
• if any bit is set in the loadMask: The curerent configuration is discarded and rebuilt from all the lower layers
Note that commands can be combined. The sequence of execution is clear, save, then load. Also note that this message is considered deprecated. Use UBX-CFG-VALSET and UBX-CFG-VALDEL with the appropriate layers instead. These new messages support selective saving and clearing to retain the behaviour removed from this message.</td></tr>
<tr><td></td><td><i>Header</i></td><td><i>Class</i></td><td><i>ID</i></td><td><i>Length (Bytes)</i></td><td><i>Payload</i></td><td><i>Checksum</i></td></tr>
</table>

Message Structure	0xB5 0x62	0x06	0x09	(12) or (13)	see below	CK_A CK_B

Payload Contents:					
Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	X4	-	`clearMask`	-	Mask for configuration to clear (see graphic below)
4	X4	-	`saveMask`	-	Mask for configuration to save (see graphic below)
8	X4	-	`loadMask`	-	Mask for configuration to load (see graphic below)
Start of optional block					
12	X1	-	`deviceMask`	-	Mask which selects the memory devices for saving and/or clearing operation Note that if a deviceMask is not provided, the receiver defaults the operation requested to Battery Backed RAM (BBR) and Flash (if available) (see graphic below)
End of optional block					

Bitfield clearMask

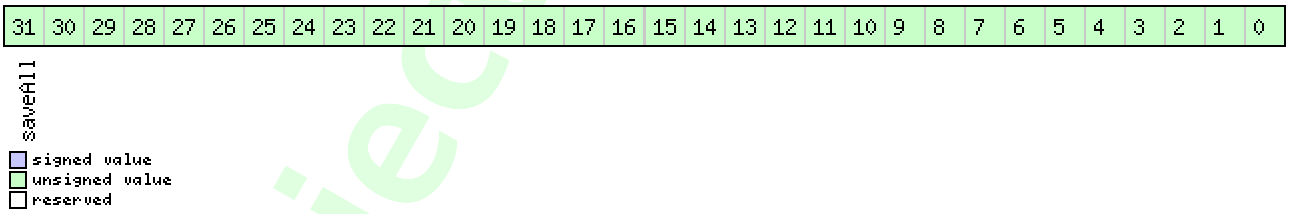
This graphic explains the bits of `clearMask`



Name	*Description*
`clearAll`	Clear all saved configuration from the selected non-volatile memory if any bit is set

Bitfield saveMask

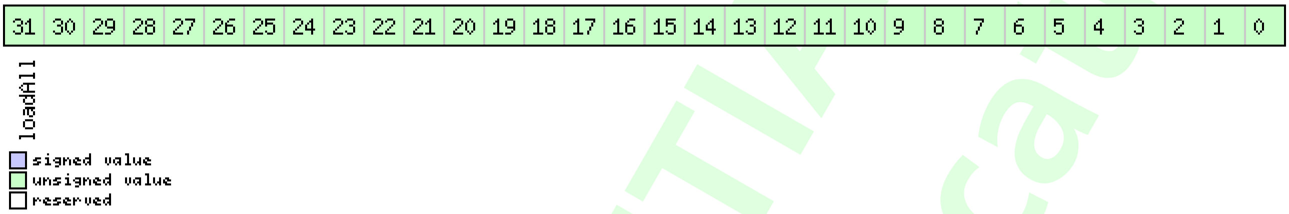
This graphic explains the bits of `saveMask`



Name	Description
saveAll	Save all current configuration to the selected non-volatile memory if any bit is set

Bitfield loadMask

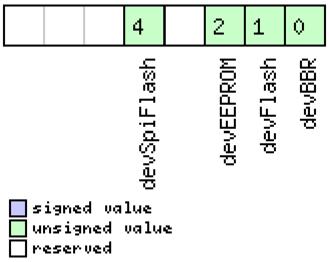
This graphic explains the bits of loadMask



Name	Description
loadAll	Discard current configuration and rebuilt it from lower non-volatile memory layers if any bit is set

Bitfield deviceMask

This graphic explains the bits of deviceMask



Name	Description
devBBR	Battery Backed RAM
devFlash	Flash
devEEPROM	EEPROM
devSpiFlash	SPI Flash

6.9.3 UBX-CFG-DAT (0x06 0x06)

6.9.3.1 Set User-defined Datum.

Message	**UBX-CFG-DAT**
Description	**Set User-defined Datum.**
Firmware	Supported on: - u-blox 9 with protocol version 27
Type	Set
Comment	**This message is deprecated in protocol versions greater than 23.01. Use UBX-CFG-VALSET, UBX-CFG-VALGET, UBX-CFG-VALDEL instead.** See the Legacy UBX Message Fields Reference for the corresponding configuration item.

	Header	Class	ID	Length (Bytes)	Payload	Checksum
Message Structure	0xB5 0x62	0x06	0x06	44	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	Number Format	Scaling	Name	Unit	Description

CFG-DAT continued

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	R8	-	`majA`	m	Semi-major Axis (accepted range = 6,300,000.0 to 6,500,000.0 meters).
8	R8	-	`flat`	-	1.0 / Flattening (accepted range is 0.0 to 500.0).
16	R4	-	`dX`	m	X Axis shift at the origin (accepted range is +/- 5000.0 meters).
20	R4	-	`dY`	m	Y Axis shift at the origin (accepted range is +/- 5000.0 meters).
24	R4	-	`dZ`	m	Z Axis shift at the origin (accepted range is +/- 5000.0 meters).
28	R4	-	`rotX`	s	Rotation about the X Axis (accepted range is +/- 20.0 milli-arc seconds).
32	R4	-	`rotY`	s	Rotation about the Y Axis (accepted range is +/- 20.0 milli-arc seconds).
36	R4	-	`rotZ`	s	Rotation about the Z Axis (accepted range is +/- 20.0 milli-arc seconds).
40	R4	-	`scale`	ppm	Scale change (accepted range is 0.0 to 50.0 parts per million).

6.9.3.2 The currently defined Datum

Message	**UBX-CFG-DAT**
Description	**The currently defined Datum**
Firmware	Supported on: • u-blox 9 with protocol version 27
Type	Get
Comment	**This message is deprecated in protocol versions greater than 23.01. Use `UBX-CFG-VALSET`, `UBX-CFG-VALGET`, `UBX-CFG-VALDEL` instead.** See the Legacy UBX Message Fields Reference for the corresponding configuration item. Returns the parameters of the currently defined datum. If no user-defined datum has been set, this will default to WGS84.

	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x06	0x06	52	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U2	-	`datumNum`	-	Datum Number: 0 = WGS84, 0xFFFF = user-defined
2	CH[6]	-	`datumName`	-	ASCII String: WGS84 or USER
8	R8	-	`majA`	m	Semi-major Axis (accepted range = 6,300,000.0 to 6,500,000.0 meters).
16	R8	-	`flat`	-	1.0 / Flattening (accepted range is 0.0 to 500.0).

CFG-DAT continued

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
24	R4	-	`dX`	m	X Axis shift at the origin (accepted range is +/- 5000.0 meters).
28	R4	-	`dY`	m	Y Axis shift at the origin (accepted range is +/- 5000.0 meters).
32	R4	-	`dZ`	m	Z Axis shift at the origin (accepted range is +/- 5000.0 meters).
36	R4	-	`rotX`	s	Rotation about the X Axis (accepted range is +/- 20.0 milli-arc seconds).
40	R4	-	`rotY`	s	Rotation about the Y Axis (accepted range is +/- 20.0 milli-arc seconds).
44	R4	-	`rotZ`	s	Rotation about the Z Axis (accepted range is +/- 20.0 milli-arc seconds).
48	R4	-	`scale`	ppm	Scale change (accepted range is 0.0 to 50.0 parts per million).

6.9.4 UBX-CFG-DGNSS (0x06 0x70)

6.9.4.1 DGNSS configuration

Message	**UBX-CFG-DGNSS**
Description	**DGNSS configuration**
Firmware	Supported on: • u-blox 9 with protocol version 27 (**only with High Precision GNSS products**)
Type	Get/Set
Comment	**This message is deprecated in protocol versions greater than 23.01. Use `UBX-CFG-VALSET`, `UBX-CFG-VALGET`, `UBX-CFG-VALDEL` instead.** See the Legacy UBX Message Fields Reference for the corresponding configuration item. This message allows the user to configure the DGNSS configuration of the receiver.

	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x06	0x70	4	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`dgnssMode`	-	Specifies differential mode: 2: RTK float: No attempts are made to fix ambiguities. 3: RTK fixed: Ambiguities are fixed whenever possible.
1	U1[3]	-	`reserved1`	-	Reserved

6.9.5 UBX-CFG-DYNSEED (0x06 0x85)

6.9.5.1 Programming the dynamic seed for the host interface signature

Message	**UBX-CFG-DYNSEED**
Description	**Programming the dynamic seed for the host interface signature**
Firmware	Supported on: • u-blox 9 with protocol version 27
Type	Set
Comment	The message can be used to program the dynamic seed for the host interface signature. If successfully configured, the message will answer with ACK, otherwise with NAK. Before the first programming, it is assumed that the dynamic seed is all '0'.

	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x06	0x85	12	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`version`	-	Message version (0x01 for this version)
1	U1[3]	-	`reserved1`	-	Reserved
4	U4	-	`seedHi`	-	high word of dynamic seed
8	U4	-	`seedLo`	-	low word of dynamic seed

6.9.6 UBX-CFG-FIXSEED (0x06 0x84)

6.9.6.1 Programming the fixed seed for host interface signature

Message	**UBX-CFG-FIXSEED**
Description	**Programming the fixed seed for host interface signature**
Firmware	Supported on: • u-blox 9 with protocol version 27
Type	Set
Comment	The message can be used to program the fixed seed for the host interface signature. Moreover it will configure the set of messages that will be signed (min. 1, max. 10). If the class ID of the message is 0 the configuration is ignored for that message. If successfully configured, the message will answer with ACK, otherwise with NAK.

	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x06	0x84	12 + 2*length	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`version`	-	Message version (0x02 for this version)
1	U1	-	`length`	-	Number of registered messages (min. 1, max. 10)
2	U1[2]	-	`reserved1`	-	Reserved
4	U4	-	`seedHi`	-	high word of fixed seed
8	U4	-	`seedLo`	-	low word of fixed seed
Start of repeated block (length times)					
12 + 2*N	U1	-	`classId`	-	Class ID on the message
13 + 2*N	U1	-	`msgId`	-	Message ID on the message

CFG-FIXSEED continued

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
End of repeated block					

6.9.7 UBX-CFG-GEOFENCE (0x06 0x69)

6.9.7.1 Geofencing configuration

Message	**UBX-CFG-GEOFENCE**
Description	**Geofencing configuration**
Firmware	Supported on: • u-blox 9 with protocol version 27
Type	Get/Set
Comment	**This message is deprecated in protocol versions greater than 23.01. Use** `UBX-CFG-VALSET`, `UBX-CFG-VALGET`, `UBX-CFG-VALDEL` **instead.** See the Legacy UBX Message Fields Reference for the corresponding configuration item. Gets or sets the geofencing configuration If the receiver is sent a valid new configuration, it will respond with a `UBX-ACK-ACK` message and immediately change to the new configuration. Otherwise the receiver will reject the request, by issuing a `UBX-ACK-NAK` and continuing operation with the previous configuration. Note that the acknowledge message does not indicate whether the PIO configuration has been successfully applied (pin assigned), it only indicates the successful configuration of the feature. The configured PIO must be previously unoccupied for successful assignment.

	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x06	0x69	8 + 12*numFences	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`version`	-	Message version (=0x00 for this version)
1	U1	-	`numFences`	-	Number of geofences contained in this message. Note that the receiver can only store a limited number of geofences (currently 4).
2	U1	-	`confLvl`	-	Required confidence level for state evaluation. This value times the position's standard deviation (sigma) defines the confidence band. 0=no confidence required, 1=68%, 2=95%, 3=99.7% etc.
3	U1[1]	-	`reserved1`	-	Reserved
4	U1	-	`pioEnabled`	-	1 = Enable PIO combined fence state output, 0 = disable
5	U1	-	`pinPolarity`	-	PIO pin polarity. 0 = Low means inside, 1 = Low means outside. Unknown state is always high.
6	U1	-	`pin`	-	PIO pin number
7	U1[1]	-	`reserved2`	-	Reserved
Start of repeated block (numFences times)					
8 + 12*N	I4	1e-7	`lat`	deg	Latitude of the geofence circle center

CFG-GEOFENCE continued

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
12 + 12*N	I4	1e-7	`lon`	deg	Longitude of the geofence circle center
16 + 12*N	U4	1e-2	`radius`	m	Radius of the geofence circle
End of repeated block					

6.9.8 UBX-CFG-GNSS (0x06 0x3E)

6.9.8.1 GNSS system configuration

Message	**UBX-CFG-GNSS**
Description	**GNSS system configuration**
Firmware	Supported on: • u-blox 9 with protocol version 27
Type	Get/Set
Comment	**This message is deprecated in protocol versions greater than 23.01. Use** `UBX-CFG-VALSET`, `UBX-CFG-VALGET`, `UBX-CFG-VALDEL` **instead.** See the Legacy UBX Message Fields Reference for the corresponding configuration item. Gets or sets the GNSS system channel sharing configuration. If the receiver is sent a valid new configuration, it will respond with a `UBX-ACK-ACK` message and immediately change to the new configuration. Otherwise the receiver will reject the request, by issuing a `UBX-ACK-NAK` and continuing operation with the previous configuration. Configuration requirements: • It is necessary for at least one major GNSS to be enabled, after applying the new configuration to the current one. • It is also required that at least 4 tracking channels are available to each enabled major GNSS, i.e. `maxTrkCh` must have a minimum value of 4 for each enabled major GNSS. • The number of tracking channels in use must not exceed the number of tracking channels available in hardware, and the sum of all reserved tracking channels needs to be less than or equal to the number of tracking channels in use. Notes: • To avoid cross-correlation issues, it is recommended that GPS and QZSS are always both enabled or both disabled. • Polling this message returns the configuration of all supported GNSS, whether enabled or not; it may also include GNSS unsupported by the particular product, but in such cases the enable flag will always be unset. • See section Satellite Numbering for a description of the GNSS IDs available. • Configuration specific to the GNSS system can be done via other messages (e.g. **UBX-CFG-SBAS**).

	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x06	0x3E	4 + 8*numConfigBlocks	*see below*	CK_A CK_B

Payload Contents:

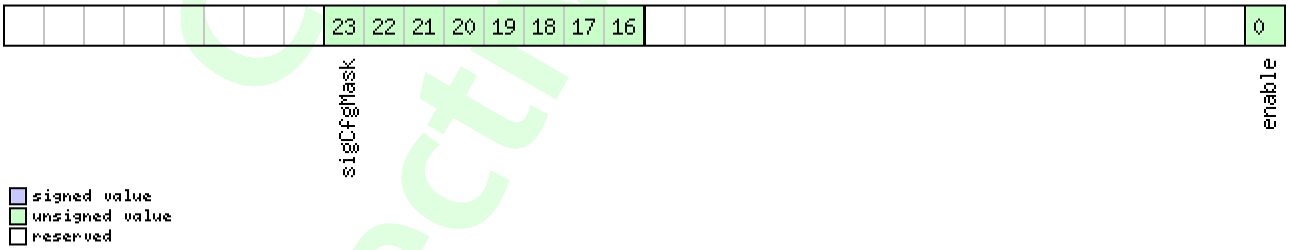
Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`msgVer`	-	Message version (=0 for this version)

CFG-GNSS continued

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
1	U1	-	`numTrkChHw`	-	Number of tracking channels available in hardware (read only)
2	U1	-	`numTrkChUse`	-	(Read only in protocol versions greater than 23) Number of tracking channels to use. Must be > 0, <= numTrkChHw. If 0xFF, then number of tracking channels to use will be set to numTrkChHw.
3	U1	-	`numConfigBlocks`	-	Number of configuration blocks following
Start of repeated block (numConfigBlocks times)					
4 + 8*N	U1	-	`gnssId`	-	System identifier (see Satellite Numbering)
5 + 8*N	U1	-	`resTrkCh`	-	(Read only in protocol versions greater than 23) Number of reserved (minimum) tracking channels for this system.
6 + 8*N	U1	-	`maxTrkCh`	-	(Read only in protocol versions greater than 23) Maximum number of tracking channels used for this system. Must be > 0, >= resTrkChn, <= numTrkChUse and <= maximum number of tracking channels supported for this system.
7 + 8*N	U1	-	`reserved1`	-	Reserved
8 + 8*N	X4	-	`flags`	-	bitfield of flags. At least one signal must be configured in every enabled system. (see graphic below)
End of repeated block					

Bitfield flags

This graphic explains the bits of `flags`



Name	Description
enable	Enable this system
sigCfgMask	Signal configuration mask When gnssId is 0 (GPS) * 0x01 = GPS L1C/A * 0x10 = GPS L2C When gnssId is 1 (SBAS) * 0x01 = SBAS L1C/A When gnssId is 2 (Galileo) * 0x01 = Galileo E1 * 0x20 = Galileo E5b When gnssId is 3 (BeiDou) * 0x01 = BeiDou B1I * 0x10 = BeiDou B2I When gnssId is 4 (IMES) * 0x01 = IMES L1 When gnssId is 5 (QZSS) * 0x01 = QZSS L1C/A * 0x04 = QZSS L1S * 0x10 = QZSS L2C When gnssId is 6 (GLONASS) * 0x01 = GLONASS L1 * 0x10 = GLONASS L2

6.9.9 UBX-CFG-INF (0x06 0x02)

6.9.9.1 Poll configuration for one protocol

Message	**UBX-CFG-INF**				
Description	**Poll configuration for one protocol**				
Firmware	Supported on: • u-blox 9 with protocol version 27				
Type	Poll Request				
Comment	**This message is deprecated in protocol versions greater than 23.01. Use UBX-CFG-VALSET, UBX-CFG-VALGET, UBX-CFG-VALDEL instead.** See the Legacy UBX Message Fields Reference for the corresponding configuration item.				
Message Structure	Header	Class	ID	Length (Bytes)	Payload	Checksum
	0xB5 0x62	0x06	0x02	1	see below	CK_A CK_B

Payload Contents:

Byte Offset	Number Format	Scaling	Name	Unit	Description
0	U1	-	protocolID	-	Protocol Identifier, identifying the output protocol for this Poll Request. The following are valid Protocol Identifiers: 0: UBX Protocol 1: NMEA Protocol 2-255: Reserved

6.9.9.2 Information message configuration

Message	**UBX-CFG-INF**
Description	**Information message configuration**
Firmware	Supported on: • u-blox 9 with protocol version 27
Type	Get/Set
Comment	**This message is deprecated in protocol versions greater than 23.01. Use `UBX-CFG-VALSET`, `UBX-CFG-VALGET`, `UBX-CFG-VALDEL` instead.** The value of infMsgMask[x] below are that each bit represents one of the INF class messages (Bit 0 for ERROR, Bit 1 for WARNING and so on.). For a complete list, see the Message Class INF. Several configurations can be concatenated to one input message. In this case the payload length can be a multiple of the normal length. Output messages from the module contain only one configuration unit. Note that I/O Ports 1 and 2 correspond to serial ports 1 and 2. I/O port 0 is DDC. I/O port 3 is USB. I/O port 4 is SPI. I/O port 5 is reserved for future use.

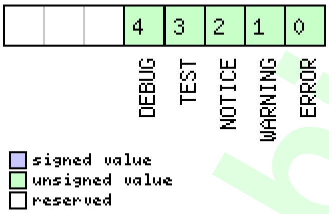
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x06	0x02	0 + 10*N	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
Start of repeated block (N times)					
N*10	U1	-	`protocolID`	-	Protocol Identifier, identifying for which protocol the configuration is set/get. The following are valid Protocol Identifiers: 0: UBX Protocol 1: NMEA Protocol 2-255: Reserved
1 + 10*N	U1[3]	-	`reserved1`	-	Reserved
4 + 10*N	X1[6]	-	`infMsgMask`	-	A bit mask, saying which information messages are enabled on each I/O port (see graphic below)
End of repeated block					

Bitfield infMsgMask

This graphic explains the bits of `infMsgMask`



Name	*Description*
`ERROR`	enable ERROR
`WARNING`	enable WARNING
`NOTICE`	enable NOTICE
`TEST`	enable TEST
`DEBUG`	enable DEBUG

6.9.10 UBX-CFG-ITFM (0x06 0x39)

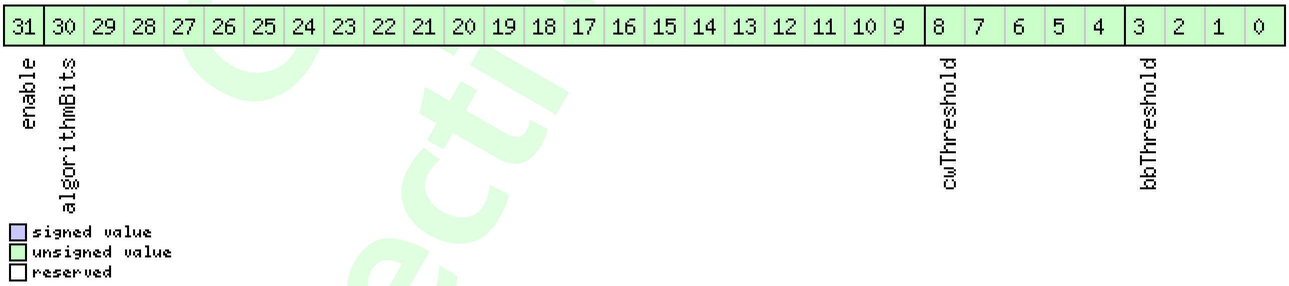
6.9.10.1 Jamming/Interference Monitor configuration

<table>
<tr><td><i>Message</i></td><td colspan="6"><b>UBX-CFG-ITFM</b></td></tr>
<tr><td><i>Description</i></td><td colspan="6"><b>Jamming/Interference Monitor configuration</b></td></tr>
<tr><td><i>Firmware</i></td><td colspan="6">Supported on:
• u-blox 9 with protocol version 27</td></tr>
<tr><td><i>Type</i></td><td colspan="6">Get/Set</td></tr>
<tr><td><i>Comment</i></td><td colspan="6"><b>This message is deprecated in protocol versions greater than 23.01. Use</b> UBX-CFG-VALSET, UBX-CFG-VALGET, UBX-CFG-VALDEL <b>instead.</b>
See the Legacy UBX Message Fields Reference for the corresponding configuration item.
Configuration of Jamming/Interference monitor.</td></tr>
<tr><td rowspan="2"><i>Message Structure</i></td><td><i>Header</i></td><td><i>Class</i></td><td><i>ID</i></td><td><i>Length (Bytes)</i></td><td><i>Payload</i></td><td><i>Checksum</i></td></tr>
<tr><td>0xB5 0x62</td><td>0x06</td><td>0x39</td><td>8</td><td><i>see below</i></td><td>CK_A CK_B</td></tr>
<tr><td colspan="7"><i>Payload Contents:</i></td></tr>
</table>

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	X4	-	`config`	-	interference config word. (see graphic below)
4	X4	-	`config2`	-	extra settings for jamming/interference monitor (see graphic below)

Bitfield config

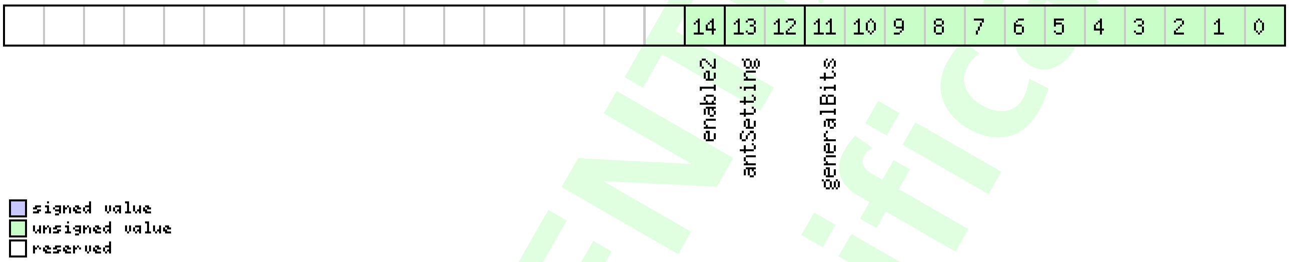
This graphic explains the bits of `config`



Name	Description
`bbThreshold`	Broadband jamming detection threshold (unit = dB)
`cwThreshold`	CW jamming detection threshold (unit = dB)
`algorithmBits`	reserved algorithm settings - should be set to 0x16B156 in hex for correct settings
`enable`	enable interference detection

Bitfield config2

This graphic explains the bits of `config2`



Name	Description
`generalBits`	general settings - should be set to 0x31E in hex for correct setting
`antSetting`	antennaSetting, 0=unknown, 1=passive, 2=active
`enable2`	Set to 1 to scan auxiliary bands (u-blox 8 / u-blox M8 only, otherwise ignored)

6.9.11 UBX-CFG-LOGFILTER (0x06 0x47)

6.9.11.1 Data Logger Configuration

Message	**UBX-CFG-LOGFILTER**
Description	**Data Logger Configuration**
Firmware	Supported on: • u-blox 9 with protocol version 27
Type	Get/Set
Comment	**This message is deprecated in protocol versions greater than 23.01. Use `UBX-CFG-VALSET`, `UBX-CFG-VALGET`, `UBX-CFG-VALDEL` instead.** See the Legacy UBX Message Fields Reference for the corresponding configuration item. This message can be used to configure the data logger, i.e. to enable/disable the log recording and to get/set the position entry filter settings. Position entries can be filtered based on time difference, position difference or current speed thresholds. Position and speed filtering also have a minimum time interval. A position is logged if any of the thresholds are exceeded. If a threshold is set to zero it is ignored. The maximum rate of position logging is 1Hz. The filter settings will be configured to the provided values only if the 'applyAllFilterSettings' flag is set. This allows the recording to be enabled/disabled independently of configuring the filter settings. It is supported to configure the data logger in the absence of a logging file. By doing so, once the logging file is created, the data logger configuration will take effect immediately and logging recording and filtering will activate according to the configuration.

	Header	Class	ID	Length (Bytes)	Payload	Checksum
Message Structure	0xB5 0x62	0x06	0x47	12	*see below*	CK_A CK_B

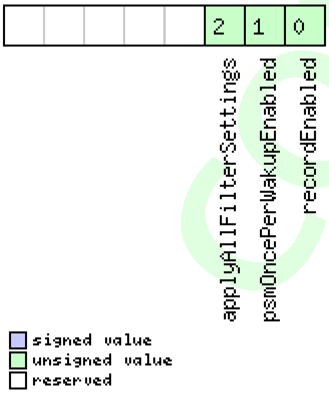
Payload Contents:

CFG-LOGFILTER continued

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`version`	-	The version of this message. Set to 1
1	X1	-	`flags`	-	Flags (see graphic below)
2	U2	-	`minInterval`	s	Minimum time interval between logged positions (0 = not set). **This is only applied in combination with the speed and/or position thresholds.** If both minInterval and timeThreshold are set, minInterval must be less than or equal to timeThreshold.
4	U2	-	`timeThreshold`	s	If the time difference is greater than the threshold then the position is logged (0 = not set).
6	U2	-	`speedThreshold`	m/s	If the current speed is greater than the threshold then the position is logged (0 = not set). minInterval also applies
8	U4	-	`positionThreshold`	m	If the 3D position difference is greater than the threshold then the position is logged (0 = not set). minInterval also applies

Bitfield flags

This graphic explains the bits of `flags`



Name	Description
recordEnabled	1 = enable recording, 0 = disable recording
psmOncePerWakupEnabled	1 = enable recording only one single position per PSM on/off mode wake-up period, 0 = disable once per wake-up
applyAllFilterSettings	1 = apply all filter settings, 0 = only apply recordEnabled

6.9.12 UBX-CFG-MSG (0x06 0x01)

6.9.12.1 Poll a message configuration

Message	**UBX-CFG-MSG**
Description	**Poll a message configuration**
Firmware	Supported on: • u-blox 9 with protocol version 27
Type	Poll Request
Comment	**This message is deprecated in protocol versions greater than 23.01. Use UBX-CFG-VALSET, UBX-CFG-VALGET, UBX-CFG-VALDEL instead.** See the Legacy UBX Message Fields Reference for the corresponding configuration item.

	Header	Class	ID	Length (Bytes)	Payload	Checksum
Message Structure	0xB5 0x62	0x06	0x01	2	see below	CK_A CK_B

Payload Contents:

Byte Offset	Number Format	Scaling	Name	Unit	Description
0	U1	-	msgClass	-	Message Class
1	U1	-	msgID	-	Message Identifier

6.9.12.2 Set Message Rate(s)

Message	**UBX-CFG-MSG**
Description	**Set Message Rate(s)**
Firmware	Supported on: • u-blox 9 with protocol version 27
Type	Get/Set
Comment	**This message is deprecated in protocol versions greater than 23.01. Use UBX-CFG-VALSET, UBX-CFG-VALGET, UBX-CFG-VALDEL instead.** See the Legacy UBX Message Fields Reference for the corresponding configuration item. Set/Get message rate configuration (s) to/from the receiver. • Send rate is relative to the event a message is registered on. For example, if the rate of a navigation message is set to 2, the message is sent every second navigation solution. For configuring NMEA messages, the section NMEA Messages Overview describes Class and Identifier numbers used.

	Header	Class	ID	Length (Bytes)	Payload	Checksum
Message Structure	0xB5 0x62	0x06	0x01	8	see below	CK_A CK_B

Payload Contents:

Byte Offset	Number Format	Scaling	Name	Unit	Description
0	U1	-	msgClass	-	Message Class

CFG-MSG continued

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
1	U1	-	`msgID`	-	Message Identifier
2	U1[6]	-	`rate`	-	Send rate on I/O Port (6 Ports)

6.9.12.3 Set Message Rate

Message	**UBX-CFG-MSG**				
Description	**Set Message Rate**				
Firmware	Supported on: • u-blox 9 with protocol version 27				
Type	Get/Set				
Comment	**This message is deprecated in protocol versions greater than 23.01. Use `UBX-CFG-VALSET`, `UBX-CFG-VALGET`, `UBX-CFG-VALDEL` instead.** See the Legacy UBX Message Fields Reference for the corresponding configuration item. Set message rate configuration for the current port.				
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x06	0x01	3	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`msgClass`	-	Message Class
1	U1	-	`msgID`	-	Message Identifier
2	U1	-	`rate`	-	Send rate on current Port

6.9.13 UBX-CFG-NAV5 (0x06 0x24)

6.9.13.1 Navigation Engine Settings

Message	**UBX-CFG-NAV5**					
Description	**Navigation Engine Settings**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Get/Set					
Comment	**This message is deprecated in protocol versions greater than 23.01. Use `UBX-CFG-VALSET`, `UBX-CFG-VALGET`, `UBX-CFG-VALDEL` instead.** See the Legacy UBX Message Fields Reference for the corresponding configuration item.					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x06	0x24	36	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	X2	-	`mask`	-	Parameters Bitmask. Only the masked parameters will be applied. (see graphic below)

CFG-NAV5 continued

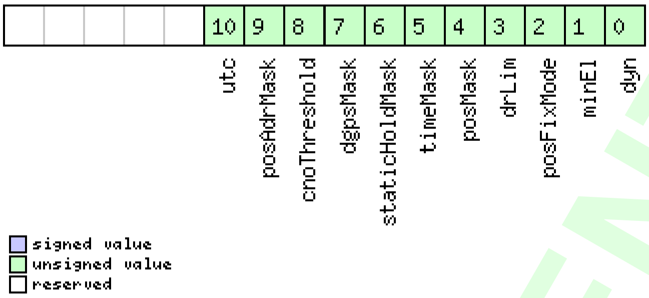
Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
2	U1	-	`dynModel`	-	Dynamic platform model: 0: portable 2: stationary 3: pedestrian 4: automotive 5: sea 6: airborne with <1g acceleration 7: airborne with <2g acceleration 8: airborne with <4g acceleration 9: wrist worn watch 10: bike
3	U1	-	`fixMode`	-	Position Fixing Mode: 1: 2D only 2: 3D only 3: auto 2D/3D
4	I4	0.01	`fixedAlt`	m	Fixed altitude (mean sea level) for 2D fix mode.
8	U4	0.0001	`fixedAltVar`	m^2	Fixed altitude variance for 2D mode.
12	I1	-	`minElev`	deg	Minimum Elevation for a GNSS satellite to be used in NAV
13	U1	-	`drLimit`	s	Reserved
14	U2	0.1	`pDop`	-	Position DOP Mask to use
16	U2	0.1	`tDop`	-	Time DOP Mask to use
18	U2	-	`pAcc`	m	Position Accuracy Mask
20	U2	-	`tAcc`	m	Time Accuracy Mask
22	U1	-	`staticHoldThresh`	cm/s	Static hold threshold
23	U1	-	`dgnssTimeout`	s	DGNSS timeout
24	U1	-	`cnoThreshNumSVs`	-	Number of satellites required to have C/N0 above cnoThresh for a fix to be attempted
25	U1	-	`cnoThresh`	dBHz	C/N0 threshold for deciding whether to attempt a fix
26	U2	-	`pAccAdr`	m	ADR position accuracy mask (only supported on the ADR product variant)
28	U2	-	`staticHoldMaxDist`	m	Static hold distance threshold (before quitting static hold)
30	U1	-	`utcStandard`	-	UTC standard to be used: 0: Automatic; receiver selects based on GNSS configuration (see GNSS time bases). 3: UTC as operated by the U.S. Naval Observatory (USNO); derived from GPS time 6: UTC as operated by the former Soviet Union; derived from GLONASS time 7: UTC as operated by the National Time Service Center, China; derived from BeiDou time

CFG-NAV5 continued

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
31	U1[5]	-	`reserved1`	-	Reserved

Bitfield mask

This graphic explains the bits of `mask`



Name	*Description*
`dyn`	Apply dynamic model settings
`minEl`	Apply minimum elevation settings
`posFixMode`	Apply fix mode settings
`drLim`	Reserved
`posMask`	Apply position mask settings
`timeMask`	Apply time mask settings
`staticHoldMask`	Apply static hold settings
`dgpsMask`	Apply DGPS settings.
`cnoThreshold`	Apply CNO threshold settings (cnoThresh, cnoThreshNumSVs).
`posAdrMask`	Apply ADR position mask settings (only supported on the ADR product variant).
`utc`	Apply UTC settings.

6.9.14 UBX-CFG-NAVX5 (0x06 0x23)

6.9.14.1 Navigation Engine Expert Settings

Message	**UBX-CFG-NAVX5**					
Description	**Navigation Engine Expert Settings**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Get/Set					
Comment	**This message is deprecated in protocol versions greater than 23.01. Use `UBX-CFG-VALSET`, `UBX-CFG-VALGET`, `UBX-CFG-VALDEL` instead.** See the Legacy UBX Message Fields Reference for the corresponding configuration item.					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x06	0x23	40	*see below*	CK_A CK_B
Payload Contents:						

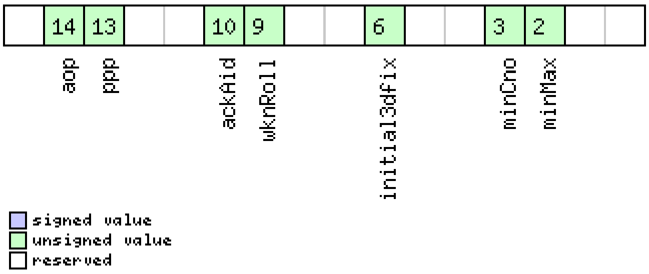
Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U2	-	`version`	-	Message version (2 for this version)

CFG-NAVX5 continued

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
2	X2	-	`mask1`	-	First parameters bitmask. Only the flagged parameters will be applied, unused bits must be set to 0. (see graphic below)
4	X4	-	`mask2`	-	Second parameters bitmask. Only the flagged parameters will be applied, unused bits must be set to 0. (see graphic below)
8	U1[2]	-	`reserved1`	-	Reserved
10	U1	-	`minSVs`	#SVs	Minimum number of satellites for navigation
11	U1	-	`maxSVs`	#SVs	Maximum number of satellites for navigation
12	U1	-	`minCNO`	dBHz	Minimum satellite signal level for navigation
13	U1	-	`reserved2`	-	Reserved
14	U1	-	`iniFix3D`	-	1 = initial fix must be 3D
15	U1[2]	-	`reserved3`	-	Reserved
17	U1	-	`ackAiding`	-	1 = issue acknowledgements for assistance message input
18	U2	-	`wknRollover`	-	GPS week rollover number; GPS week numbers will be set correctly from this week up to 1024 weeks after this week. Setting this to 0 reverts to firmware default.
20	U1	-	`sigAttenCompMode`	dBHz	*Only supported on certain products*
21	U1	-	`reserved4`	-	Reserved
22	U1[2]	-	`reserved5`	-	Reserved
24	U1[2]	-	`reserved6`	-	Reserved
26	U1	-	`usePPP`	-	1 = use Precise Point Positioning (only available with the PPP product variant)
27	U1	-	`aopCfg`	-	*AssistNow Autonomous* configuration (see graphic below)
28	U1[2]	-	`reserved7`	-	Reserved
30	U2	-	`aopOrbMaxErr`	m	Maximum acceptable (modeled) *AssistNow Autonomous* orbit error (valid range = 5..1000, or 0 = reset to firmware default)
32	U1[4]	-	`reserved8`	-	Reserved
36	U1[3]	-	`reserved9`	-	Reserved
39	U1	-	`useAdr`	-	*Only supported on certain products*

Bitfield mask1

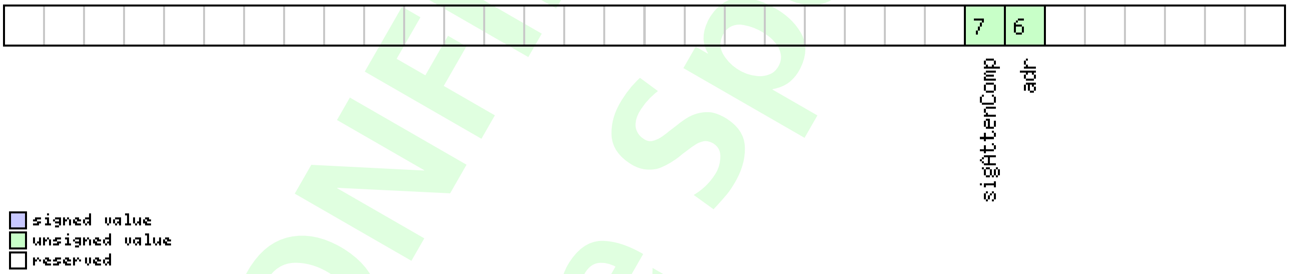
This graphic explains the bits of `mask1`



Name	Description
minMax	1 = apply min/max SVs settings
minCno	1 = apply minimum C/N0 setting
initial3dfix	1 = apply initial 3D fix settings
wknRoll	1 = apply GPS weeknumber rollover settings
ackAid	1 = apply assistance acknowledgement settings
ppp	1 = apply usePPP flag
aop	1 = apply aopCfg (useAOP flag) and aopOrbMaxErr settings (AssistNow Autonomous)

Bitfield mask2

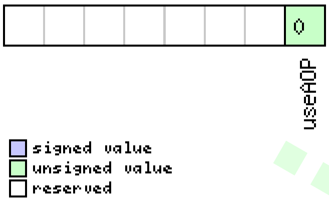
This graphic explains the bits of `mask2`



Name	Description
adr	Apply ADR/UDR sensor fusion on/off setting (useAdr flag)
sigAttenComp	*Only supported on certain products*

Bitfield aopCfg

This graphic explains the bits of `aopCfg`



Name	*Description*
`useAOP`	1 = enable *AssistNow Autonomous*

6.9.15 UBX-CFG-NMEA (0x06 0x17)

6.9.15.1 Extended NMEA protocol configuration V1

Message	**UBX-CFG-NMEA**					
Description	**Extended NMEA protocol configuration V1**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Get/Set					
Comment	**This message is deprecated in protocol versions greater than 23.01. Use `UBX-CFG-VALSET`, `UBX-CFG-VALGET`, `UBX-CFG-VALDEL` instead.** Set/Get the NMEA protocol configuration. See section NMEA Protocol Configuration for a detailed description of the configuration effects on NMEA output. See the Legacy UBX Message Fields Reference for the corresponding configuration item.					
Message Structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xB5 0x62	0x06	0x17	20	*see below*	CK_A CK_B

Payload Contents:

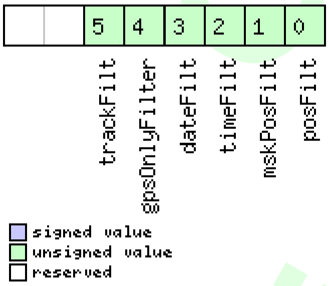
Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	X1	-	`filter`	-	filter flags (see graphic below)
1	U1	-	`nmeaVersion`	-	0x41: NMEA version 4.1 0x40: NMEA version 4.0 0x23: NMEA version 2.3 0x21: NMEA version 2.1
2	U1	-	`numSV`	-	Maximum Number of SVs to report per TalkerId. 0: unlimited 8: 8 SVs 12: 12 SVs 16: 16 SVs
3	X1	-	`flags`	-	flags (see graphic below)
4	X4	-	`gnssToFilter`	-	Filters out satellites based on their GNSS. If a bitfield is enabled, the corresponding satellites will be not output. (see graphic below)
8	U1	-	`svNumbering`	-	Configures the display of satellites that do not have an NMEA-defined value. Note: this does not apply to satellites with an unknown ID. 0: Strict - Satellites are not output 1: Extended - Use proprietary numbering (see Satellite Numbering)

CFG-NMEA continued

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
9	U1	-	`mainTalkerId`	-	By default the main Talker ID (i.e. the Talker ID used for all messages other than GSV) is determined by the GNSS assignment of the receiver's channels (see `UBX-CFG-GNSS`). This field enables the main Talker ID to be overridden. 0: Main Talker ID is not overridden 1: Set main Talker ID to 'GP' 2: Set main Talker ID to 'GL' 3: Set main Talker ID to 'GN' 4: Set main Talker ID to 'GA' 5: Set main Talker ID to 'GB'
10	U1	-	`gsvTalkerId`	-	By default the Talker ID for GSV messages is GNSS specific (as defined by NMEA). This field enables the GSV Talker ID to be overridden. 0: Use GNSS specific Talker ID (as defined by NMEA) 1: Use the main Talker ID
11	U1	-	`version`	-	Message version (set to 1 for this version)
12	CH[2]	-	`bdsTalkerId`	-	Sets the two characters that should be used for the BeiDou Talker ID If these are set to zero, the default BeiDou TalkerId will be used
14	U1[6]	-	`reserved1`	-	Reserved

Bitfield filter

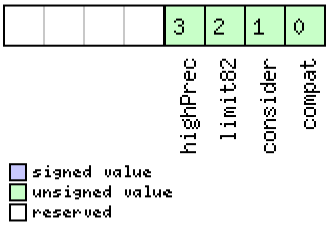
This graphic explains the bits of `filter`



Name	Description
posFilt	Enable position output for failed or invalid fixes
mskPosFilt	Enable position output for invalid fixes
timeFilt	Enable time output for invalid times
dateFilt	Enable date output for invalid dates
gpsOnlyFilter	Restrict output to GPS satellites only
trackFilt	Enable COG output even if COG is frozen

Bitfield flags

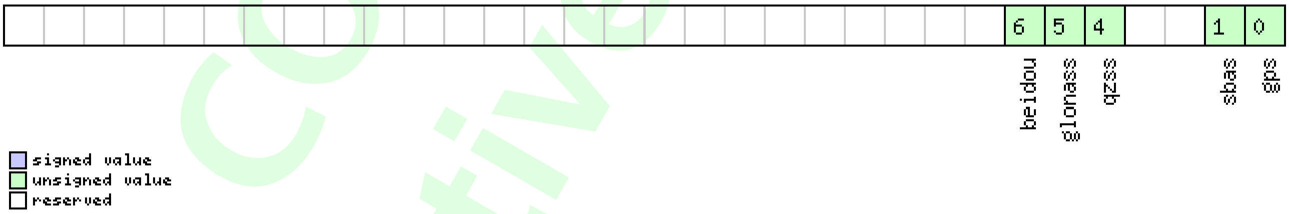
This graphic explains the bits of `flags`



Name	Description
compat	enable compatibility mode. This might be needed for certain applications when customer's NMEA parser expects a fixed number of digits in position coordinates
consider	enable considering mode.
limit82	enable strict limit to 82 characters maximum.
highPrec	enable high precision mode. This flag cannot be set in conjunction with either Compatibility Mode or Limit82 Mode.

Bitfield gnssToFilter

This graphic explains the bits of `gnssToFilter`



Name	Description
gps	Disable reporting of GPS satellites
sbas	Disable reporting of SBAS satellites
qzss	Disable reporting of QZSS satellites
glonass	Disable reporting of GLONASS satellites
beidou	Disable reporting of BeiDou satellites

6.9.16 UBX-CFG-ODO (0x06 0x1E)

6.9.16.1 Odometer, Low-speed COG Engine Settings

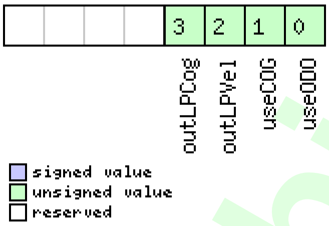
Message	**UBX-CFG-ODO**					
Description	**Odometer, Low-speed COG Engine Settings**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Get/Set					
Comment	**This feature is not supported for the FTS product variant.** **This message is deprecated in protocol versions greater than 23.01. Use** UBX-CFG-VALSET, UBX-CFG-VALGET, UBX-CFG-VALDEL **instead.** See the Legacy UBX Message Fields Reference for the corresponding configuration item.					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x06	0x1E	20	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`version`	-	Message version (0 for this version)
1	U1[3]	-	`reserved1`	-	Reserved
4	U1	-	`flags`	-	*Odometer/Low-speed COG filter* flags (see graphic below)
5	X1	-	`odoCfg`	-	*Odometer filter* settings (see graphic below)
6	U1[6]	-	`reserved2`	-	Reserved
12	U1	1e-1	`cogMaxSpeed`	m/s	Speed below which course-over-ground (COG) is computed with the low-speed COG filter
13	U1	-	`cogMaxPosAcc`	m	Maximum acceptable position accuracy for computing COG with the low-speed COG filter
14	U1[2]	-	`reserved3`	-	Reserved
16	U1	-	`velLpGain`	-	Velocity low-pass filter level, range 0..255
17	U1	-	`cogLpGain`	-	COG low-pass filter level (at speed < 8 m/s), range 0..255
18	U1[2]	-	`reserved4`	-	Reserved

Bitfield flags

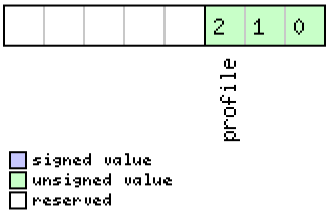
This graphic explains the bits of `flags`



Name	*Description*
`useODO`	Odometer enabled flag
`useCOG`	Low-speed COG filter enabled flag
`outLPVel`	Output low-pass filtered velocity flag
`outLPCog`	Output low-pass filtered heading (COG) flag

Bitfield odoCfg

This graphic explains the bits of `odoCfg`



Name	*Description*
`profile`	Profile type (0=running, 1=cycling, 2=swimming, 3=car, 4=custom)

6.9.17 UBX-CFG-OTP (0x06 0x41)

6.9.17.1 Poll OTP content

Message	**UBX-CFG-OTP**					
Description	**Poll OTP content**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Poll Request					
Comment	-					
Message Structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xB5 0x62	0x06	0x41	0	*see below*	CK_A CK_B
No payload						

6.9.17.2 OTP content

Message	**UBX-CFG-OTP**					
Description	**OTP content**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Get					
Comment	The content of the OTP, returned as a result of a POLL message. See section eFuse for a detailed description of the OTP content.					
Message Structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xB5 0x62	0x06	0x41	128	*see below*	CK_A CK_B
Payload Contents:						

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1[128]	-	`otpContent`	-	The content of the OTP

6.9.17.3 Writes OTP content

Message	**UBX-CFG-OTP**
Description	**Writes OTP content**
Firmware	Supported on: • u-blox 9 with protocol version 27
Type	Set
Comment	Writes content to the OTP. An ACK will be returned when the content was successfully written. A NAK when there was a problem.

	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x06	0x41	0 + 1*N	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
Start of repeated block (N times)					
N*1	U1	-	`payload`	-	Payload. Use u-center to compose the message.
End of repeated block					

6.9.17.4 Write the USB vendor ID file (0x20)

Message	**UBX-CFG-OTP**
Description	**Write the USB vendor ID file (0x20)**
Firmware	Supported on: • u-blox 9 with protocol version 27
Type	Set
Comment	Writes content to the OTP. An ACK will be returned when the content was successfully written. A NAK when there was a problem.

	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x06	0x41	10	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`version`	-	Message version (0x01 for this version)
1	U1	-	`operation`	-	Operation (0x01 for this operation)
2	U1	-	`fileID`	-	File identifier (0x20 for this file)
3	U1	-	`length`	-	File length and validity (0x82 for this file)
4	U1[4]	-	`reserved1`	-	Reserved
8	U2	-	`vendorId`	-	The USB vendor ID that overwrites the default.

6.9.17.5 Write the USB vendor string file (0x21)

Message	**UBX-CFG-OTP**
Description	**Write the USB vendor string file (0x21)**
Firmware	Supported on: • u-blox 9 with protocol version 27
Type	Set
Comment	Writes content to the OTP. An ACK will be returned when the content was successfully written. A NAK when there was a problem.

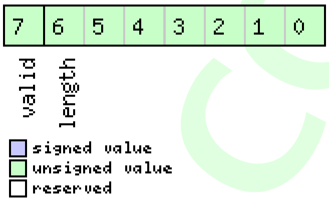
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x06	0x41	8 + 1*length	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`version`	-	Message version (0x01 for this version)
1	U1	-	`operation`	-	Operation (0x01 for this operation)
2	U1	-	`fileID`	-	File identifier (0x21 for this file)
3	X1	-	`length_mask`	-	The length of the USB vendor string. (see graphic below)
4	U1[4]	-	`reserved1`	-	Reserved
Start of repeated block (length times)					
8 + 1*N	U1	-	`string`	-	The USB vendor string that overwrites the default.
End of repeated block					

Bitfield length_mask

This graphic explains the bits of `length_mask`



Name	*Description*
`length`	The length of the USB vendor string (max. 32)
`valid`	Validity: set to 0x1

6.9.17.6 Write the USB product ID file (0x22)

Message	**UBX-CFG-OTP**
Description	**Write the USB product ID file (0x22)**
Firmware	Supported on: • u-blox 9 with protocol version 27
Type	Set
Comment	Writes content to the OTP. An ACK will be returned when the content was successfully written. A NAK when there was a problem.

	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x06	0x41	10	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`version`	-	Message version (0x01 for this version)
1	U1	-	`operation`	-	Operation (0x01 for this operation)
2	U1	-	`fileID`	-	File identifier (0x22 for this file)
3	U1	-	`length`	-	File length and validity (0x82 for this file)
4	U1[4]	-	`reserved1`	-	Reserved
8	U2	-	`vendorId`	-	The USB product ID that overwrites the default.

6.9.17.7 Write the USB product string file (0x23)

Message	**UBX-CFG-OTP**
Description	**Write the USB product string file (0x23)**
Firmware	Supported on: • u-blox 9 with protocol version 27
Type	Set
Comment	Writes content to the OTP. An ACK will be returned when the content was successfully written. A NAK when there was a problem.

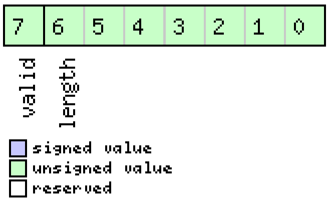
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x06	0x41	8 + 1*length	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`version`	-	Message version (0x01 for this version)
1	U1	-	`operation`	-	Operation (0x01 for this operation)
2	U1	-	`fileID`	-	File identifier (0x23 for this file)
3	X1	-	`length_mask`	-	The length of the USB product string (see graphic below)
4	U1[4]	-	`reserved1`	-	Reserved
Start of repeated block (length times)					
8 + 1*N	U1	-	`string`	-	The USB product string that overwrites the default.
End of repeated block					

Bitfield length_mask

This graphic explains the bits of `length_mask`



Name	Description
`length`	The length of the USB product string (max. 32)
`valid`	Validity: set to 0x1

6.9.17.8 Write the receiver configuration file (0x30)

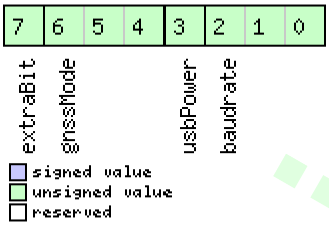
Message	**UBX-CFG-OTP**					
Description	**Write the receiver configuration file (0x30)**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Set					
Comment	Writes content to the OTP. An ACK will be returned when the content was successfully written. A NAK when there was a problem.					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x06	0x41	9	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`version`	-	Message version (0x01 for this version)
1	U1	-	`operation`	-	Operation (0x01 for this operation)
2	U1	-	`fileID`	-	File identifier (0x30 for this file)
3	U1	-	`length`	-	File length and validity (0x81 for this file)
4	U1[4]	-	`reserved1`	-	Reserved
8	X1	-	`config`	-	The receiver configuration. (see graphic below)

Bitfield config

This graphic explains the bits of `config`



Name	Description

`baudrate`	Baud rate of UART port
	`0x7 = 9600 (default)`
	`0x6 = 1200`
	`0x5 = 2400`
	`0x4 = 4800`
	`0x3 = 19200`
	`0x2 = 38400`
	`0x1 = 57600`
	`0x0 = 115200`

`usbPower`	USB power setting
	`0x1 = self powered (default)`
	`0x0 = bus powered`

`gnssMode`	GNSS mode
	`0x7 = continuous mode (default)`
	`all other values are reserved`

`extraBit`	Extra bit: set to 1

6.9.17.9 Write the post production flags file (0x32)

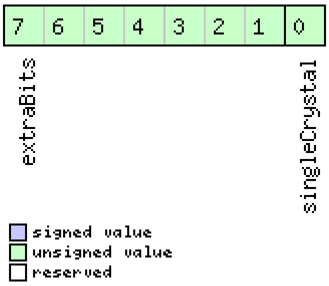
Message	**UBX-CFG-OTP**				
Description	**Write the post production flags file (0x32)**				
Firmware	Supported on: • u-blox 9 with protocol version 27				
Type	Set				
Comment	Writes content to the OTP. An ACK will be returned when the content was successfully written. A NAK when there was a problem.				
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x06	0x41	9	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`version`	-	Message version (0x01 for this version)
1	U1	-	`operation`	-	Operation (0x01 for this operation)
2	U1	-	`fileID`	-	File identifier (0x32 for this file)
3	U1	-	`length`	-	File length and validity (0x81 for this file)
4	U1[4]	-	`reserved1`	-	Reserved
8	X1	-	`config`	-	System selection. (see graphic below)

Bitfield config

This graphic explains the bits of `config`



Name	Description

`singleCrystal`	Single crystal mode
	`0x1 = enable`
	`0x0 = disable`

`extraBits`	Extra bits: set to 0x7F

6.9.17.10 Write the oscillator offset calibration file (0x36)

Message	**UBX-CFG-OTP**
Description	**Write the oscillator offset calibration file (0x36)**
Firmware	Supported on: - u-blox 9 with protocol version 27
Type	Set
Comment	Writes content to the OTP. An ACK will be returned when the content was successfully written. A NAK when there was a problem.

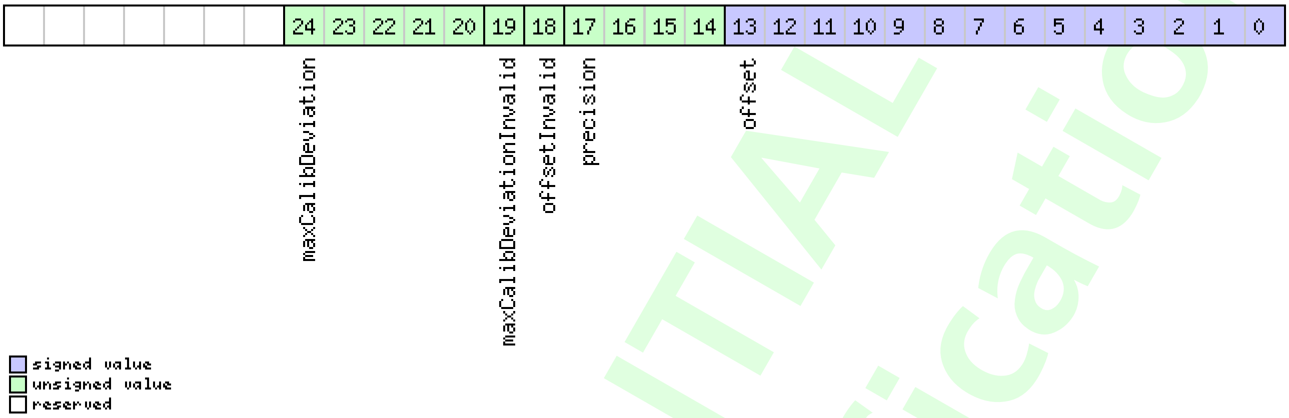
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x06	0x41	13	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`version`	-	Message version (0x01 for this version)
1	U1	-	`operation`	-	Operation (0x01 for this operation)
2	U1	-	`fileID`	-	File identifier (0x36 for this file)
3	U1	-	`length`	-	File length and validity (0x85 for this file)
4	U1[4]	-	`reserved1`	-	Reserved
8	X4	-	`oscillatorOffsetCalibration1`	-	Oscillator offset calibration (see graphic below)
12	U1	-	`extraByte`	-	Extra byte: set to 0xFF

Bitfield oscillatorOffsetCalibration1

This graphic explains the bits of `oscillatorOffsetCalibration1`



Name	Description
`offset`	Offset of the oscillator [0.1ppm]
`precision`	Precision of the offset [1ppm]
`offsetInvalid`	Flag to indicate if the offset (and precision) is invalid
`maxCalibDeviationInvalid`	Flag to indicate if maxCalibDeviation is invalid
`maxCalibDeviation`	maximum calibration deviation [1ppm]

6.9.18 UBX-CFG-PIO (0x06 0x2c)

6.9.18.1 Set PIO pins

Message	**UBX-CFG-PIO**					
Description	**Set PIO pins**					
Firmware	Supported on: - u-blox 9 with protocol version 27					
Type	Set					
Comment	PIO setup for production testing. Request 1 (SET-PIN) will change to a PIO test state and set a desired pin state for PIO pins (except those specified as 'unaltered'). Clearly, PIO settings could disturb system operation or, possibly, even cause damage. Request 0 (EXIT_TEST) will end test state and restore the PIO state back as it was before the most recent change into test state. Requests 2 (EXTERNAL-NODRIVE) and 3 (EXTERNAL-DRIVE) explore the external connection state of pins and automatically trigger a `UBX-MON-PIO` message containing the results of that examination. EXTERNAL-NODRIVE does not hard drive outputs, while request EXTERNAL-DRIVE does this - and could in principle cause damage by driving a pin which is externally being driven the other way. However, the test time is limited, the PIOs are only capable of driving a limited current and only one pin is tested at a time to minimize the possibility of this. An EXIT-TEST message only contans the request, while the other requests contain a value per pin. For a SET-PIN message, this determines the PIO state that will be set. For EXTERNAL-NODRIVE and EXTERNAL-DRIVE messages the value should be 6 for pins to be examined or 0 for pins which are not to be examined.					
Message Structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xB5 0x62	0x06	0x2c	19	*see below*	CK_A CK_B

Payload Contents:					
Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`version`	-	Message version (0 for this version)
1	U1	-	`request`	-	Possible requests: 0: EXIT-TEST. No test, back to normal operation 1: SET-PIN. Set pin state (unaltered, float, high, low, pull-up) 2: EXTERNAL-NODRIVE. Find external connection state of pins with requiredPinstate 6 as far as is possible without driving pins 3: EXTERNAL_DRIVE. Find external connection state of pins with requiredPinstate 6 as far as possible, allowed to drive pins
Start of repeated block (17 times)					
2 + 1*N	U1	-	`requiredPinState`	-	One value per PIO pin: 0: Pin state unaltered 1: Set pin floating 2: Pin pulled down 3: Pin pulled up 4: Pin driven high 5: Pin driven low 6: Investigate this pin
End of repeated block					

6.9.19 UBX-CFG-PRT (0x06 0x00)

6.9.19.1 Polls the configuration for one I/O Port

Message	**UBX-CFG-PRT**					
Description	**Polls the configuration for one I/O Port**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Poll Request					
Comment	**This message is deprecated in protocol versions greater than 23.01. Use `UBX-CFG-VALSET`, `UBX-CFG-VALGET`, `UBX-CFG-VALDEL` instead.** See the Legacy UBX Message Fields Reference for the corresponding configuration item. Sending this message with a port ID as payload results in having the receiver return the configuration for the specified port.					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x06	0x00	1	*see below*	CK_A CK_B

Payload Contents:					
Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`PortID`	-	Port Identifier Number (see the other versions of CFG-PRT for valid values)

6.9.19.2 Port Configuration for UART

Message	**UBX-CFG-PRT**
Description	**Port Configuration for UART**
Firmware	Supported on: • u-blox 9 with protocol version 27
Type	Get/Set
Comment	**This message is deprecated in protocol versions greater than 23.01. Use** `UBX-CFG-VALSET`, `UBX-CFG-VALGET`, `UBX-CFG-VALDEL` **instead.** See the Legacy UBX Message Fields Reference for the corresponding configuration item. Several configurations can be concatenated to one input message. In this case the payload length can be a multiple of the normal length (see the other versions of CFG-PRT). Output messages from the module contain only one configuration unit. Note that this message can affect baud rate and other transmission parameters. Because there may be messages queued for transmission there may be uncertainty about which protocol applies to such messages. In addition a message currently in transmission may be corrupted by a protocol change. Host data reception parameters may have to be changed to be able to receive future messages, including the acknowledge message resulting from the CFG-PRT message.

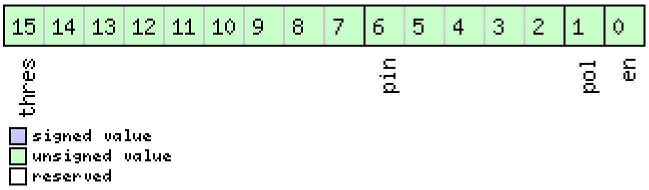
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x06	0x00	20	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`portID`	-	Port Identifier Number (see Integration Manual for valid UART port IDs)
1	U1	-	`reserved1`	-	Reserved
2	X2	-	`txReady`	-	TX ready PIN configuration (see graphic below)
4	X4	-	`mode`	-	A bit mask describing the UART mode (see graphic below)
8	U4	-	`baudRate`	Bits/s	Baud rate in bits/second
12	X2	-	`inProtoMask`	-	A mask describing which input protocols are active. Each bit of this mask is used for a protocol. Through that, multiple protocols can be defined on a single port. (see graphic below)
14	X2	-	`outProtoMask`	-	A mask describing which output protocols are active. Each bit of this mask is used for a protocol. Through that, multiple protocols can be defined on a single port. (see graphic below)
16	X2	-	`flags`	-	Flags bit mask (see graphic below)
18	U1[2]	-	`reserved2`	-	Reserved

Bitfield txReady

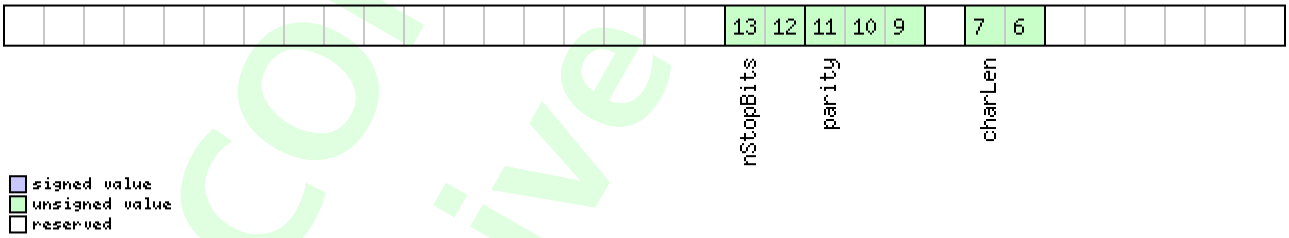
This graphic explains the bits of `txReady`



Name	Description
en	Enable TX ready feature for this port
pol	Polarity 0 High-active 1 Low-active
pin	PIO to be used (must not be in use already by another function)
thres	Threshold The given threshold is multiplied by 8 bytes. The TX ready PIN goes active after >= thres*8 bytes are pending for the port and going inactive after the last pending bytes have been written to hardware (0-4 bytes before end of stream). 0x000 no threshold 0x001 8byte 0x002 16byte ... 0x1FE 4080byte 0x1FF 4088byte

Bitfield mode

This graphic explains the bits of `mode`



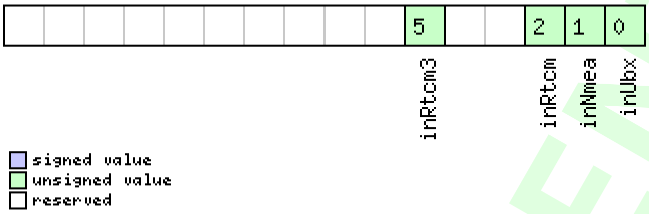
Name	Description
charLen	Character Length 00 5bit (not supported) 01 6bit (not supported) 10 7bit (supported only with parity) 11 8bit
parity	000 Even Parity 001 Odd Parity 10X No Parity X1X Reserved

Bitfield mode Description continued

Name	*Description*
`nStopBits`	Number of Stop Bits 00 1 Stop Bit 01 1.5 Stop Bit 10 2 Stop Bit 11 0.5 Stop Bit

Bitfield inProtoMask

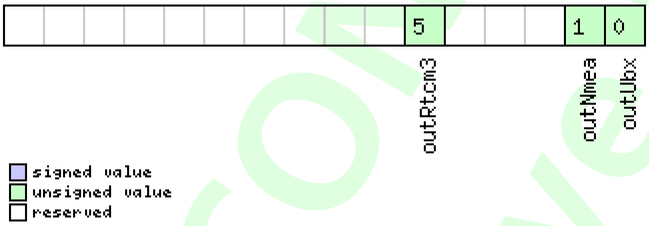
This graphic explains the bits of `inProtoMask`



Name	*Description*
`inUbx`	UBX protocol
`inNmea`	NMEA protocol
`inRtcm`	RTCM2 protocol
`inRtcm3`	RTCM3 protocol

Bitfield outProtoMask

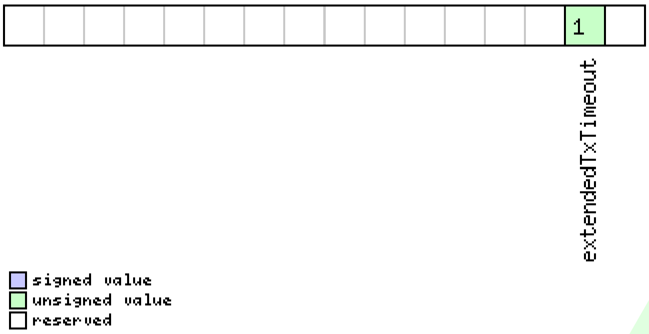
This graphic explains the bits of `outProtoMask`



Name	*Description*
`outUbx`	UBX protocol
`outNmea`	NMEA protocol
`outRtcm3`	RTCM3 protocol

Bitfield flags

This graphic explains the bits of `flags`



Name	*Description*
`extendedTxTimeout`	Extended TX timeout: if set, the port will timeout if allocated TX memory >=4 kB and no activity for 1.5s. If not set the port will timeout if no activity for 1.5s regardless on the amount of allocated TX memory.

6.9.19.3 Port Configuration for USB Port

Message	**UBX-CFG-PRT**					
Description	**Port Configuration for USB Port**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Get/Set					
Comment	**This message is deprecated in protocol versions greater than 23.01. Use `UBX-CFG-VALSET`, `UBX-CFG-VALGET`, `UBX-CFG-VALDEL` instead.** See the Legacy UBX Message Fields Reference for the corresponding configuration item. Several configurations can be concatenated to one input message. In this case the payload length can be a multiple of the normal length (see the other versions of CFG-PRT). Output messages from the module contain only one configuration unit.					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x06	0x00	20	*see below*	CK_A CK_B

Payload Contents:

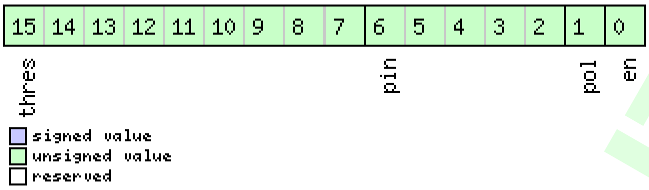
Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`portID`	-	Port Identifier Number (= 3 for USB port)
1	U1	-	`reserved1`	-	Reserved
2	X2	-	`txReady`	-	TX ready PIN configuration (see graphic below)
4	U1[8]	-	`reserved2`	-	Reserved
12	X2	-	`inProtoMask`	-	A mask describing which input protocols are active. Each bit of this mask is used for a protocol. Through that, multiple protocols can be defined on a single port. (see graphic below)
14	X2	-	`outProtoMask`	-	A mask describing which output protocols are active. Each bit of this mask is used for a protocol. Through that, multiple protocols can be defined on a single port. (see graphic below)
16	U1[2]	-	`reserved3`	-	Reserved

CFG-PRT continued

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
18	U1[2]	-	`reserved4`	-	Reserved

Bitfield txReady

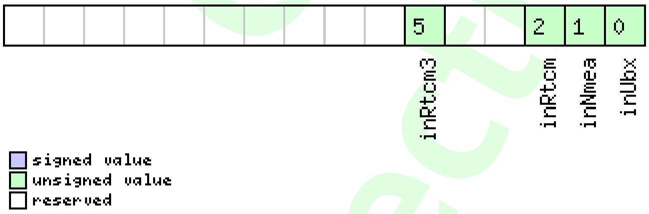
This graphic explains the bits of `txReady`



Name	*Description*
`en`	Enable TX ready feature for this port
`pol`	Polarity 0 High-active 1 Low-active
`pin`	PIO to be used (must not be in use already by another function)
`thres`	Threshold The given threshold is multiplied by 8 bytes. The TX ready PIN goes active after >= thres*8 bytes are pending for the port and going inactive after the last pending bytes have been written to hardware (0-4 bytes before end of stream). 0x000 no threshold 0x001 8byte 0x002 16byte ... 0x1FE 4080byte 0x1FF 4088byte

Bitfield inProtoMask

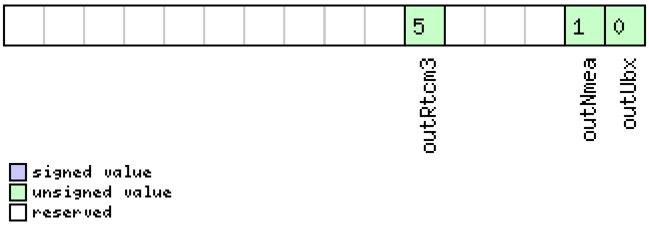
This graphic explains the bits of `inProtoMask`



Name	*Description*
`inUbx`	UBX protocol
`inNmea`	NMEA protocol
`inRtcm`	RTCM2 protocol
`inRtcm3`	RTCM3 protocol

Bitfield outProtoMask

This graphic explains the bits of `outProtoMask`



Name	Description
`outUbx`	UBX protocol
`outNmea`	NMEA protocol
`outRtcm3`	RTCM3 protocol

6.9.19.4 Port Configuration for SPI Port

Message	**UBX-CFG-PRT**
Description	**Port Configuration for SPI Port**
Firmware	Supported on: • u-blox 9 with protocol version 27
Type	Get/Set
Comment	**This message is deprecated in protocol versions greater than 23.01. Use `UBX-CFG-VALSET`, `UBX-CFG-VALGET`, `UBX-CFG-VALDEL` instead.** See the Legacy UBX Message Fields Reference for the corresponding configuration item. Several configurations can be concatenated to one input message. In this case the payload length can be a multiple of the normal length (see the other versions of CFG-PRT). Output messages from the module contain only one configuration unit.

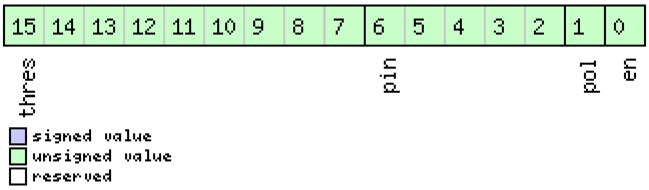
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x06	0x00	20	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`portID`	-	Port Identifier Number (= 4 for SPI port)
1	U1	-	`reserved1`	-	Reserved
2	X2	-	`txReady`	-	TX ready PIN configuration (see graphic below)
4	X4	-	`mode`	-	SPI Mode Flags (see graphic below)
8	U1[4]	-	`reserved2`	-	Reserved
12	X2	-	`inProtoMask`	-	A mask describing which input protocols are active. Each bit of this mask is used for a protocol. Through that, multiple protocols can be defined on a single port. (see graphic below)
14	X2	-	`outProtoMask`	-	A mask describing which output protocols are active. Each bit of this mask is used for a protocol. Through that, multiple protocols can be defined on a single port. (see graphic below)
16	X2	-	`flags`	-	Flags bit mask (see graphic below)
18	U1[2]	-	`reserved3`	-	Reserved

Bitfield txReady

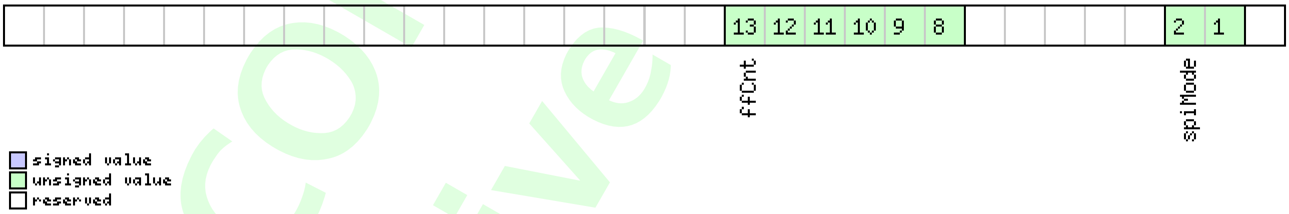
This graphic explains the bits of `txReady`



Name	Description
`en`	Enable TX ready feature for this port
`pol`	Polarity 0 High-active 1 Low-active
`pin`	PIO to be used (must not be in use already by another function)
`thres`	Threshold The given threshold is multiplied by 8 bytes. The TX ready PIN goes active after >= thres*8 bytes are pending for the port and going inactive after the last pending bytes have been written to hardware (0-4 bytes before end of stream). 0x000 no threshold 0x001 8byte 0x002 16byte ... 0x1FE 4080byte 0x1FF 4088byte

Bitfield mode

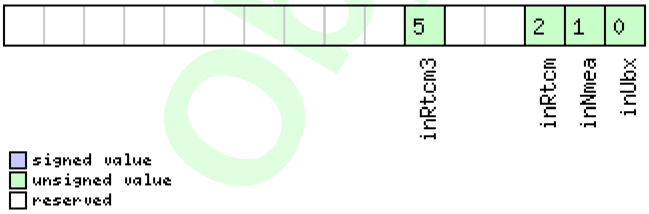
This graphic explains the bits of `mode`



Name	Description
`spiMode`	00 SPI Mode 0: CPOL = 0, CPHA = 0 01 SPI Mode 1: CPOL = 0, CPHA = 1 10 SPI Mode 2: CPOL = 1, CPHA = 0 11 SPI Mode 3: CPOL = 1, CPHA = 1
`ffCnt`	Number of bytes containing 0xFF to receive before switching off reception. Range: 0(mechanism off)-63

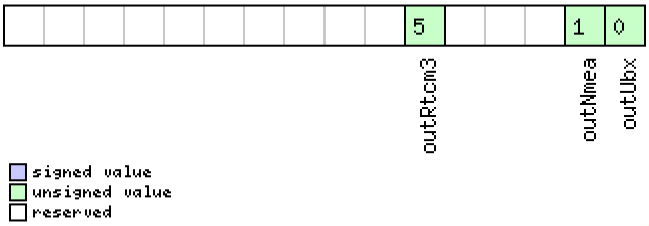
Bitfield inProtoMask

This graphic explains the bits of `inProtoMask`



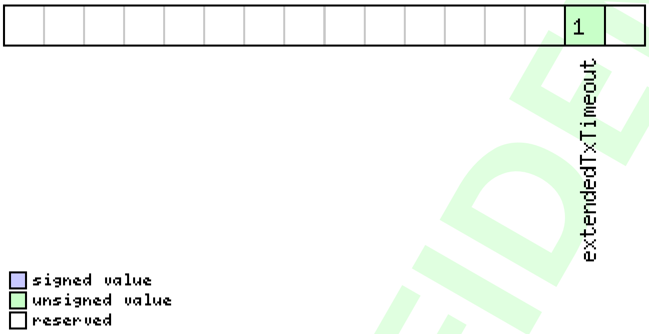
Bitfield outProtoMask

This graphic explains the bits of `outProtoMask`



Bitfield flags

This graphic explains the bits of `flags`



Name	*Description*
`extendedTxTimeout`	Extended TX timeout: if set, the port will timeout if allocated TX memory >=4 kB and no activity for 1.5s.

6.9.19.5 Port Configuration for DDC Port

Message	**UBX-CFG-PRT**					
Description	**Port Configuration for DDC Port**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Get/Set					
Comment	**This message is deprecated in protocol versions greater than 23.01. Use `UBX-CFG-VALSET`, `UBX-CFG-VALGET`, `UBX-CFG-VALDEL` instead.** See the Legacy UBX Message Fields Reference for the corresponding configuration item. Several configurations can be concatenated to one input message. In this case the payload length can be a multiple of the normal length (see the other versions of CFG-PRT). Output messages from the module contain only one configuration unit.					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x06	0x00	20	*see below*	CK_A CK_B
Payload Contents:						

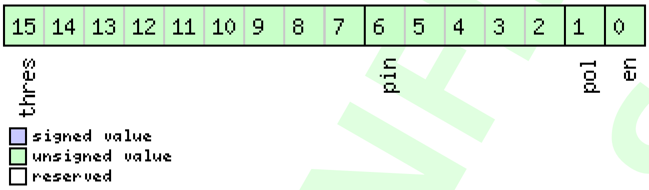
Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`portID`	-	Port Identifier Number (= 0 for DDC port)
1	U1	-	`reserved1`	-	Reserved
2	X2	-	`txReady`	-	TX ready PIN configuration (see graphic below)

CFG-PRT continued

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
4	X4	-	`mode`	-	DDC Mode Flags (see graphic below)
8	U1[4]	-	`reserved2`	-	Reserved
12	X2	-	`inProtoMask`	-	A mask describing which input protocols are active. Each bit of this mask is used for a protocol. Through that, multiple protocols can be defined on a single port. (see graphic below)
14	X2	-	`outProtoMask`	-	A mask describing which output protocols are active. Each bit of this mask is used for a protocol. Through that, multiple protocols can be defined on a single port. (see graphic below)
16	X2	-	`flags`	-	Flags bit mask (see graphic below)
18	U1[2]	-	`reserved3`	-	Reserved

Bitfield txReady

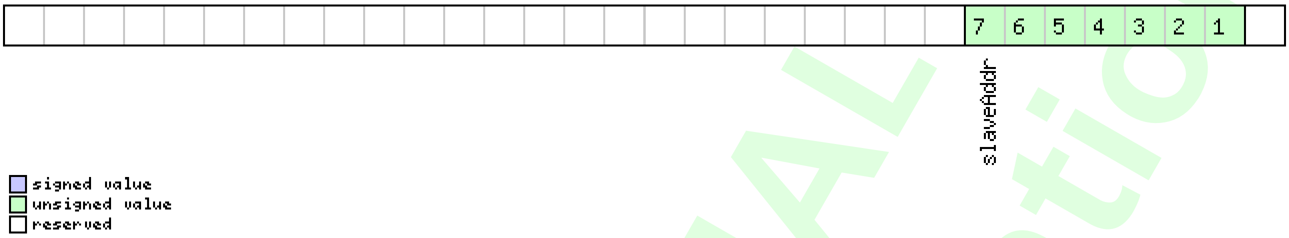
This graphic explains the bits of `txReady`



Name	*Description*
`en`	Enable TX ready feature for this port
`pol`	Polarity 0 High-active 1 Low-active
`pin`	PIO to be used (must not be in use already by another function)
`thres`	Threshold The given threshold is multiplied by 8 bytes. The TX ready PIN goes active after >= thres*8 bytes are pending for the port and going inactive after the last pending bytes have been written to hardware (0-4 bytes before end of stream). 0x000 no threshold 0x001 8byte 0x002 16byte ... 0x1FE 4080byte 0x1FF 4088byte

Bitfield mode

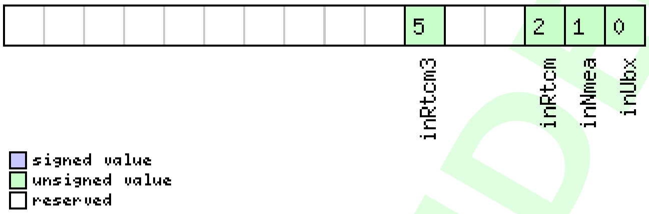
This graphic explains the bits of `mode`



Name	*Description*
`slaveAddr`	Slave address Range: 0x07 < slaveAddr < 0x78. Bit 0 must be 0

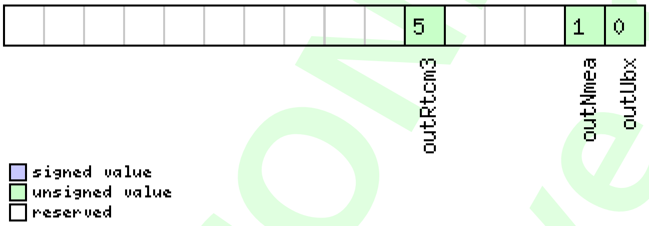
Bitfield inProtoMask

This graphic explains the bits of `inProtoMask`



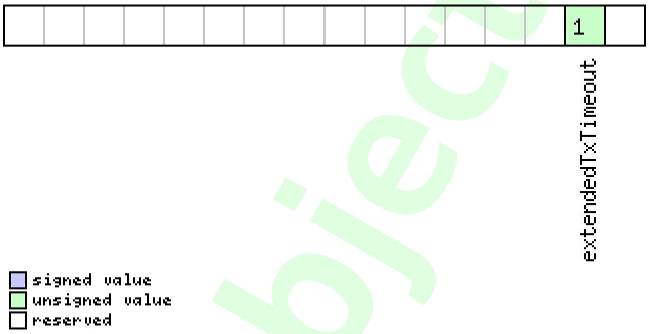
Bitfield outProtoMask

This graphic explains the bits of `outProtoMask`



Bitfield flags

This graphic explains the bits of `flags`



Name	Description
extendedTxTimeout	Extended TX timeout: if set, the port will timeout if allocated TX memory >=4 kB and no activity for 1.5s.

6.9.20 UBX-CFG-PT2 (0x06 0x59)

6.9.20.1 Production test configuration

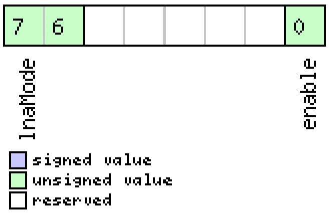
Message	**UBX-CFG-PT2**				
Description	**Production test configuration**				
Firmware	Supported on: • u-blox 9 with protocol version 27				
Type	Set				
Comment	Activate and set config for Production test mode. This allows setting a variable number of SV signal descriptors (no more than the number of RF channels of the receiver). Activating also enables output message UBX-MON-PT2.				
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x06	0x59	12 + 4*N	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	version	-	Message version (0 for this version)
1	X1	-	activate	-	production testmode (see graphic below)
2	U1	-	extint	-	input pin for the optional frequency aiding 0x00=EXTINT0 0x01=EXTINT1 0xFF=no frequency aiding
3	U1	-	reAcqCno	dBHz	C/N0 threshold to force re-acquisition (=0 means inactive)
4	U4	-	refFreq	Hz	Reference frequency
8	U4	-	refFreqAcc	ppb	Reference frequency accuracy
Start of repeated block (N times)					
12 + 4*N	U1	-	gnssId	-	GNSS identifier (see Satellite Numbering)
13 + 4*N	U1	-	svId	-	Satellite identifier (see Satellite Numbering)
14 + 4*N	U1	-	sigId	-	Signal identifier. 0 is the only value currently supported.
15 + 4*N	U1	-	accsId	-	Access identifier, used to select frequency channel in range (0-13) for GLONASS (0 = -7, 1 = -6, ..., 12 = +5, 13 = +6). Ignored for all other GNSS.
End of repeated block					

Bitfield activate

This graphic explains the bits of `activate`



Name	Description
`enable`	1=activate testmode, 0=deactivate testmode
`lnaMode`	Internal LNA Mode 0x00=Use default internal LNA settings 0x01=Switch internal LNA off 0x02=Switch internal LNA on

6.9.21 UBX-CFG-PWR (0x06 0x57)

6.9.21.1 Put receiver in a defined power state.

Message	**UBX-CFG-PWR**					
Description	**Put receiver in a defined power state.**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Set					
Comment	**This message is deprecated in protocol versions greater than 17. Use `UBX-CFG-RST` for GNSS start/stop and `UBX-RXM-PMREQ` for software backup.** See the Legacy UBX Message Fields Reference for the corresponding configuration item.					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x06	0x57	8	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`version`	-	Message version (1 for this version)
1	U1[3]	-	`reserved1`	-	Reserved
4	U4	-	`state`	-	Enter system state 0x52554E20: GNSS running 0x53544F50: GNSS stopped 0x42434B50: Software Backup. USB interface will be disabled, other wakeup source is needed.

6.9.22 UBX-CFG-RATE (0x06 0x08)

6.9.22.1 Navigation/Measurement Rate Settings

Message	**UBX-CFG-RATE**
Description	**Navigation/Measurement Rate Settings**
Firmware	Supported on: • u-blox 9 with protocol version 27
Type	Get/Set
Comment	**This feature is not supported for the FTS product variant.** **This message is deprecated in protocol versions greater than 23.01. Use `UBX-CFG-VALSET`, `UBX-CFG-VALGET`, `UBX-CFG-VALDEL` instead.** See the Legacy UBX Message Fields Reference for the corresponding configuration item. This message allows the user to alter the rate at which navigation solutions (and the measurements that they depend on) are generated by the receiver. The calculation of the navigation solution will always be aligned to the top of a second zero (first second of the week) of the configured reference time system. (Navigation period is an integer multiple of the measurement period in protocol versions greater than 17) • Each measurement triggers the measurements generation and raw data output. • The navRate value defines that every nth measurement triggers a navigation epoch. • The update rate has a direct influence on the power consumption. The more fixes that are required, the more CPU power and communication resources are required. • For most applications a 1 Hz update rate would be sufficient. • When using Power Save Mode, measurement and navigation rate can differ from the values configured here.

	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x06	0x08	6	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U2	-	`measRate`	ms	The elapsed time between GNSS measurements, which defines the rate, e.g. 100ms => 10Hz, 1000ms => 1Hz, 10000ms => 0.1Hz. Measurement rate should be greater than or equal to 25 ms.
2	U2	-	`navRate`	cycles	The ratio between the number of measurements and the number of navigation solutions, e.g. 5 means five measurements for every navigation solution. Maximum value is 127.
4	U2	-	`timeRef`	-	The time system to which measurements are aligned: 0: UTC time 1: GPS time 2: GLONASS time 3: BeiDou time 4: Galileo time

6.9.23 UBX-CFG-RINV (0x06 0x34)

6.9.23.1 Contents of Remote Inventory

Message	**UBX-CFG-RINV**
Description	**Contents of Remote Inventory**
Firmware	Supported on: • u-blox 9 with protocol version 27
Type	Get/Set
Comment	**This message is deprecated in protocol versions greater than 23.01. Use** `UBX-CFG-VALSET`, `UBX-CFG-VALGET`, `UBX-CFG-VALDEL` **instead.** If *N* is greater than 30, the excess bytes are discarded. See the Legacy UBX Message Fields Reference for the corresponding configuration item.

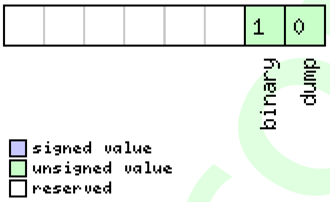
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x06	0x34	1 + 1*N	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	X1	-	`flags`	-	Flags (see graphic below)
Start of repeated block (N times)					
1 + 1*N	U1	-	`data`	-	Data to store/stored in Remote Inventory.
End of repeated block					

Bitfield flags

This graphic explains the bits of `flags`



Name	*Description*
`dump`	Dump data at startup. Does not work if flag `binary` is set.
`binary`	Data is binary.

6.9.24 UBX-CFG-RST (0x06 0x04)

6.9.24.1 Reset Receiver / Clear Backup Data Structures

Message	**UBX-CFG-RST**
Description	**Reset Receiver / Clear Backup Data Structures**
Firmware	Supported on: • u-blox 9 with protocol version 27
Type	Command
Comment	Don't expect this message to be acknowledged by the receiver. • Newer FW version won't acknowledge this message at all. • Older FW version will acknowledge this message but the acknowledge may not be sent completely before the receiver is reset.

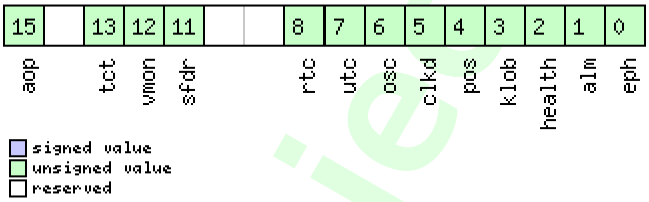
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x06	0x04	4	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	X2	-	`navBbrMask`	-	BBR Sections to clear. The following Special Sets apply: 0x0000 Hot start 0x0001 Warm start 0xFFFF Cold start (see graphic below)
2	U1	-	`resetMode`	-	Reset Type 0x00 - Hardware reset (Watchdog) immediately 0x01 - Controlled Software reset 0x02 - Controlled Software reset (GNSS only) 0x04 - Hardware reset (Watchdog) after shutdown 0x08 - Controlled GNSS stop 0x09 - Controlled GNSS start
3	U1	-	`reserved1`	-	Reserved

Bitfield navBbrMask

This graphic explains the bits of `navBbrMask`



Name	Description
eph	Ephemeris
alm	Almanac
health	Health
klob	Klobuchar parameters
pos	Position
clkd	Clock Drift
osc	Oscillator Parameter
utc	UTC Correction + GPS Leap Seconds Parameters
rtc	RTC
sfdr	SFDR Parameters (only available on the ADR product variant)
vmon	SFDR Vehicle Monitoring Parameter (only available on the ADR product variant)
tct	TCT Parameters (only available on the ADR product variant)
aop	Autonomous Orbit Parameters

6.9.25 UBX-CFG-TMODE3 (0x06 0x71)

6.9.25.1 Time Mode Settings 3

Message	**UBX-CFG-TMODE3**
Description	**Time Mode Settings 3**
Firmware	Supported on: • u-blox 9 with protocol version 27 (**only with High Precision GNSS products**)
Type	Get/Set
Comment	**This message is deprecated in protocol versions greater than 23.01. Use UBX-CFG-VALSET, UBX-CFG-VALGET, UBX-CFG-VALDEL instead.** See the Legacy UBX Message Fields Reference for the corresponding configuration item. Configures the receiver to be in Time Mode. The position referred to in this message is that of the Antenna Reference Point (ARP).

	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x06	0x71	40	*see below*	CK_A CK_B

Payload Contents:

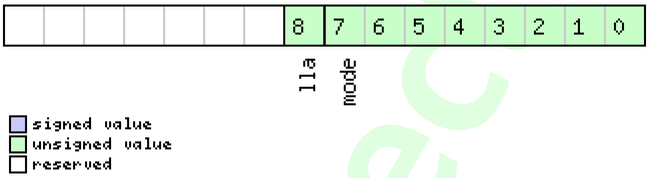
Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	version	-	Message version (0x00 for this version)
1	U1	-	reserved1	-	Reserved
2	X2	-	flags	-	Receiver mode flags (see graphic below)
4	I4	-	ecefXOrLat	cm_or_deg*1e-7	WGS84 ECEF X coordinate (or latitude) of the ARP position, depending on flags above
8	I4	-	ecefYOrLon	cm_or_deg*1e-7	WGS84 ECEF Y coordinate (or longitude) of the ARP position, depending on flags above
12	I4	-	ecefZOrAlt	cm	WGS84 ECEF Z coordinate (or altitude) of the ARP position, depending on flags above

CFG-TMODE3 continued

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
16	I1	-	`ecefXOrLatHP`	0.1_mm_or_deg*1e-9	High-precision WGS84 ECEF X coordinate (or latitude) of the ARP position, depending on flags above. Must be in the range -99..+99. The precise WGS84 ECEF X coordinate in units of cm, or the precise WGS84 ECEF latitude in units of 1e-7 degrees, is given by ecefXOrLat + (ecefXOrLatHP * 1e-2)
17	I1	-	`ecefYOrLonHP`	0.1_mm_or_deg*1e-9	High-precision WGS84 ECEF Y coordinate (or longitude) of the ARP position, depending on flags above. Must be in the range -99..+99. The precise WGS84 ECEF Y coordinate in units of cm, or the precise WGS84 ECEF longitude in units of 1e-7 degrees, is given by ecefYOrLon + (ecefYOrLonHP * 1e-2)
18	I1	-	`ecefZOrAltHP`	0.1_mm	High-precision WGS84 ECEF Z coordinate (or altitude) of the ARP position, depending on flags above. Must be in the range -99..+99. The precise WGS84 ECEF Z coordinate, or altitude coordinate, in units of cm is given by ecefZOrAlt + (ecefZOrAltHP * 1e-2)
19	U1	-	`reserved2`	-	Reserved
20	U4	-	`fixedPosAcc`	0.1_mm	Fixed position 3D accuracy
24	U4	-	`svinMinDur`	s	Survey-in minimum duration
28	U4	-	`svinAccLimit`	0.1_mm	Survey-in position accuracy limit
32	U1[8]	-	`reserved3`	-	Reserved

Bitfield flags

This graphic explains the bits of `flags`



Name	*Description*
`mode`	Receiver Mode: 0 Disabled 1 Survey In 2 Fixed Mode (true ARP position information required) 3-255 Reserved
`lla`	Position is given in LAT/LON/ALT (default is ECEF)

6.9.26 UBX-CFG-TP5 (0x06 0x31)

6.9.26.1 Time Pulse Parameters

Message	**UBX-CFG-TP5**
Description	**Time Pulse Parameters**
Firmware	Supported on: • u-blox 9 with protocol version 27
Type	Get/Set
Comment	This message is used to get/set time pulse parameters. For more information see section Time pulse.

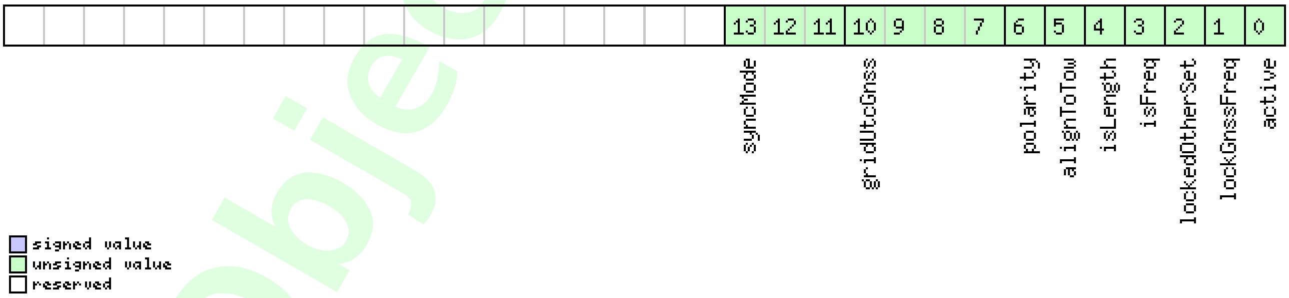
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x06	0x31	32	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`tpIdx`	-	Time pulse selection (0 = TIMEPULSE, 1 = TIMEPULSE2)
1	U1	-	`version`	-	Message version (0x01 for this version)
2	U1[2]	-	`reserved1`	-	Reserved
4	I2	-	`antCableDelay`	ns	Antenna cable delay
6	I2	-	`rfGroupDelay`	ns	RF group delay
8	U4	-	`freqPeriod`	Hz_or_us	Frequency or period time, depending on setting of bit 'isFreq'
12	U4	-	`freqPeriodLock`	Hz_or_us	Frequency or period time when locked to GNSS time, only used if 'lockedOtherSet' is set
16	U4	-	`pulseLenRatio`	us_or_2^-32	Pulse length or duty cycle, depending on 'isLength'
20	U4	-	`pulseLenRatioLock`	us_or_2^-32	Pulse length or duty cycle when locked to GNSS time, only used if 'lockedOtherSet' is set
24	I4	-	`userConfigDelay`	ns	User configurable time pulse delay
28	X4	-	`flags`	-	Configuration flags (see graphic below)

Bitfield flags

This graphic explains the bits of `flags`



Name	*Description*
`active`	If set enable time pulse; if pin assigned to another function, other function takes precedence. Must be set for FTS variant.
`lockGnssFreq`	If set synchronize time pulse to GNSS as soon as GNSS time is valid. If not set, or before GNSS time is valid use local clock. This flag is ignored by the FTS product variant; in this case the receiver always locks to the best available time/frequency reference (which is not necessarily GNSS).
`lockedOtherSet`	If set the receiver switches between the timepulse settings given by 'freqPeriodLocked' & 'pulseLenLocked' and those given by 'freqPeriod' & 'pulseLen'. The 'Locked' settings are used where the receiver has an accurate sense of time. For non-FTS products, this occurs when GNSS solution with a reliable time is available, but for FTS products the setting syncMode field governs behavior. In all cases, the receiver only uses 'freqPeriod' & 'pulseLen' when the flag is unset.
`isFreq`	If set 'freqPeriodLock' and 'freqPeriod' are interpreted as frequency, otherwise interpreted as period.
`isLength`	If set 'pulseLenRatioLock' and 'pulseLenRatio' interpreted as pulse length, otherwise interpreted as duty cycle.
`alignToTow`	Align pulse to top of second (period time must be integer fraction of 1s). Also set 'lockGnssFreq' to use this feature. This flag is ignored by the FTS product variant; it is assumed to be always set (as is lockGnssFreq). Set maxSlewRate and maxPhaseCorrRate fields of **UBX-CFG-SMGR** to 0 to disable alignment.
`polarity`	Pulse polarity: 0: falling edge at top of second 1: rising edge at top of second
`gridUtcGnss`	Timegrid to use: 0: UTC 1: GPS 2: GLONASS 3: BeiDou 4: Galileo This flag is only relevant if 'lockGnssFreq' and 'alignToTow' are set. Note that configured GNSS time is estimated by the receiver if locked to any GNSS system. If the receiver has a valid GNSS fix it will attempt to steer the TP to the specified time grid even if the specified time is not based on information from the constellation's satellites. To ensure timing based purely on a given GNSS, restrict the supported constellations in `UBX-CFG-GNSS`.
`syncMode`	Sync Manager lock mode to use: 0: switch to 'freqPeriodLock' and 'pulseLenRatioLock' as soon as Sync Manager has an accurate time, never switch back to 'freqPeriod' and 'pulseLenRatio' 1: switch to 'freqPeriodLock' and 'pulseLenRatioLock' as soon as Sync Manager has an accurate time, and switch back to 'freqPeriod' and 'pulseLenRatio' as soon as time gets inaccurate This field is only relevant for the FTS product variant. This field is only relevant if the flag 'lockedOtherSet' is set.

6.9.27 UBX-CFG-USBTEST (0x06 0x58)

6.9.27.1 USB Testing

Message	**UBX-CFG-USBTEST**					
Description	**USB Testing**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Set					
Comment	-					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x06	0x58	2	*see below*	CK_A CK_B
Payload Contents:						

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`version`	-	Message version (0 for this version)
1	U1	-	`usbPinState`	-	Define the USB test state and output 0: Test mode disabled, normal pin usage 1: Set tristate: DP=Z DM=Z 2: Output DP=1 DM=0 3: Output DP=0 DM=1 4: Output DP=0 DM=0

6.9.28 UBX-CFG-USB (0x06 0x1B)

6.9.28.1 USB Configuration

Message	**UBX-CFG-USB**					
Description	**USB Configuration**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Get/Set					
Comment	**This message is deprecated in protocol versions greater than 23.01. Use `UBX-CFG-VALSET`, `UBX-CFG-VALGET`, `UBX-CFG-VALDEL` instead.** See the Legacy UBX Message Fields Reference for the corresponding configuration item.					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x06	0x1B	108	*see below*	CK_A CK_B
Payload Contents:						

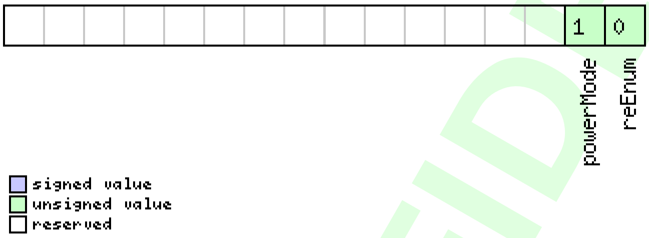
Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U2	-	`vendorID`	-	Vendor ID. This field shall only be set to registered Vendor IDs. Changing this field requires special Host drivers.
2	U2	-	`productID`	-	Product ID. Changing this field requires special Host drivers.
4	U1[2]	-	`reserved1`	-	Reserved
6	U1[2]	-	`reserved2`	-	Reserved
8	U2	-	`powerConsumption`	mA	Power consumed by the device

CFG-USB continued

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
10	X2	-	`flags`	-	various configuration flags (see graphic below)
12	CH[32]	-	`vendorString`	-	String containing the vendor name. 32 ASCII bytes including 0-termination.
44	CH[32]	-	`productString`	-	String containing the product name. 32 ASCII bytes including 0-termination.
76	CH[32]	-	`serialNumber`	-	String containing the serial number. 32 ASCII bytes including 0-termination. Changing the String fields requires special Host drivers.

Bitfield flags

This graphic explains the bits of `flags`



Name	*Description*
`reEnum`	force re-enumeration
`powerMode`	self-powered (1), bus-powered (0)

6.9.29 UBX-CFG-VALDEL (0x06 0x8C)

6.9.29.1 Deletes values corresponding to provided keys

Message	**UBX-CFG-VALDEL**
Description	**Deletes values corresponding to provided keys**
Firmware	Supported on: • u-blox 9 with protocol version 27
Type	Set
Comment	Overview: • This message can be used to delete saved configuration to effectively revert them to defaults. • This message can delete saved configuration from the Flash configuration layer and the BBR configuration layer. The changes won't be effective until these layers are loaded into the RAM layer. • This message is limited to containing a maximum of 64 keys up for deletion; i.e. N is a maximum of 64. • This message can be used multiple times and every time the result will be applied immediately. To send this message multiple times with the result being applied at the end, see version 1 of `UBX-CFG-VALDEL` that supports transactions. • This message does not check if the resulting configuration is valid. • See Receiver Configuration for details. This message returns a UBX-ACK-NAK and no configuration is applied:

	- if any key is unknown to the receiver FW - if the layers bitfield does not specify a layer to delete a value from Notes: - If a key is sent multiple times within the same message, then the value is effectively deleted only once. - Attempting to delete items that have not been set before, or that have already been deleted, is considered a valid request

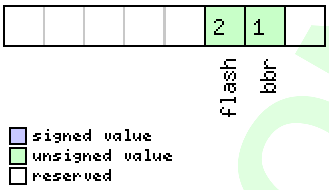
	Header	Class	ID	Length (Bytes)	Payload	Checksum
Message Structure	0xB5 0x62	0x06	0x8C	4 + 4*N	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`version`	-	Message version, set to 0
1	X1	-	`layers`	-	The layers where the configuration should be deleted from (see graphic below)
2	U1[2]	-	`reserved1`	-	Reserved
Start of repeated block (N times)					
4 + 4*N	U4	-	`keys`	-	Configuration Item IDs of the Configuration Items to be deleted
End of repeated block					

Bitfield layers

This graphic explains the bits of `layers`



Name	*Description*
`bbr`	Delete configuration from the BBR layer
`flash`	Delete configuration from the Flash layer

6.9.29.2 Deletes values corresponding to provided keys within a transaction

Message	**UBX-CFG-VALDEL**
Description	**Deletes values corresponding to provided keys within a transaction**
Firmware	Supported on: - u-blox 9 with protocol version 27
Type	Set
Comment	Overview: - This message can be used to delete saved configuration to effectively revert them to defaults. - This message can delete saved configuration from the Flash configuration layer and the BBR configuration layer. The changes won't be effective until these layers are loaded into the RAM layer. - This message is limited to containing a maximum of 64 keys up for deletion; i.e. N is a

maximum of 64.
- This message can be used multiple times with the result being managed within a transaction.
- This message does not check if the resulting configuration is valid.
- See Receiver Configuration for details.
- See version 0 of `UBX-CFG-VALDEL` for simplified version of this message.

This message returns a UBX-ACK-NAK, cancels any started transaction, and no configuration is applied:
- if any key within a transaction is unknown to the receiver FW
- if an invalid transaction state transition is requested
- if the layers bitfield changes within a transaction
- if the layers bitfield does not specify a layer to delete a value from

Notes:
- Any request for another UBX-CFG- message type (including UBX-CFG-VALSET and UBX-CFG-VALGET) will cancel any started transaction, and no configuration is applied.
- This message can be sent with no keys to delete for the purposes of managing the transaction state transition.
- If a key is sent multiple times within the same message or within the same transaction, then the value is effectively deleted only once.
- Attempting to delete items that have not been set before, or that have already been deleted, is considered a valid request

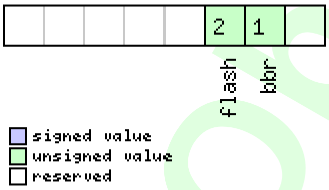
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x06	0x8C	4 + 4*N	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`version`	-	Message version, set to 1
1	X1	-	`layers`	-	The layers where the configuration should be deleted from (see graphic below)
2	X1	-	`transaction`	-	Transaction action to be applied: (see graphic below)
3	U1	-	`reserved1`	-	Reserved
Start of repeated block (N times)					
4 + 4*N	U4	-	`keys`	-	Configuration Item IDs of the Configuration Items to be deleted
End of repeated block					

Bitfield layers

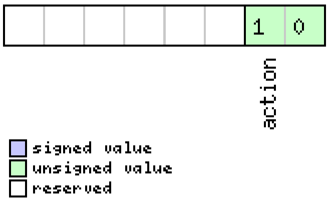
This graphic explains the bits of `layers`



Name	Description
`bbr`	Delete configuration from the BBR layer
`flash`	Delete configuration from the Flash layer

Bitfield transaction

This graphic explains the bits of `transaction`



Name	Description
`action`	Transaction action to be applied: 0: Transactionless UBX-CFG-VALDEL: In the next UBX-CFG-VALDEL, it can be either 0 or 1. If a transaction has not yet been started, the incoming configuration is applied. If a transaction has already been started, cancels any started transaction and the incoming configuration is applied. 1: (Re)Start deletion transaction: In the next UBX-CFG-VALDEL, it can be either 0, 1, 2 or 3. If a transaction has not yet been started, a transaction will be started. If a transaction has already been started, restarts the transaction, effectively removing all previous non-applied UBX-CFG-VALDEL messages. 2: Deletion transaction ongoing: In the next UBX-CFG-VALDEL, it can be either 0, 1, 2 or 3. 3: Apply and end a deletion transaction: In the next UBX-CFG-VALDEL, it can be either 0 or 1.

6.9.30 UBX-CFG-VALGET (0x06 0x8B)

6.9.30.1 Get Configuration Items

Message	**UBX-CFG-VALGET**					
Description	**Get Configuration Items**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Poll Request					
Comment	This message is used to read configuration items from the receiver. It returns the configuration data for the specified items and layer. A UBX-CFG-NAK message is returned in case one or more items are unknown to the receiver or when the number of requested items is greater than 64. Otherwise a UBX-CFG-ACK message is returned. The configuration items are identified by their configuration key IDs. See Receiver Configuration for details.					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x06	0x8B	4 + 4*N	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`version`	-	message version, set to 0

CFG-VALGET continued

Byte Offset	Number Format	Scaling	Name	Unit	Description
1	U1	-	`layer`	-	The layers from which the configuration items should be retrieved: 0 - RAM layer 1 - BBR layer 2 - Flash layer 3 - Image layer 4 - OTP layer 5 - Pin layer 6 - ROM layer 7 - Default layer
2	U1[2]	-	`reserved1`	-	Reserved
Start of repeated block (N times)					
4 + 4*N	U4	-	`keys`	-	configuration key ID selected for retrieval
End of repeated block					

6.9.30.2 Configuration Items

Message	**UBX-CFG-VALGET**					
Description	**Configuration Items**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	polled					
Comment	This message is output by the receiver to return requested configuration data (key and value pairs). See Receiver Configuration for details.					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x06	0x8B	4 + 1*N	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	Number Format	Scaling	Name	Unit	Description
0	U1	-	`version`	-	message version, set to 1
1	U1	-	`layer`	-	The layers from which the configuration items originate: 0 - RAM layer 1 - BBR 2 - Flash 3 - Image layer 4 - OTP layer 5 - Pin layer 6 - ROM layer 7 - Default
2	U1[2]	-	`reserved1`	-	Reserved
Start of repeated block (N times)					
4 + 1*N	U1	-	`cfgData`	-	configuration data (key and value pairs)

CFG-VALGET continued

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
End of repeated block					

6.9.31 UBX-CFG-VALSET (0x06 0x8A)

6.9.31.1 Sets values corresponding to provided key-value pairs

Message	**UBX-CFG-VALSET**
Description	**Sets values corresponding to provided key-value pairs**
Firmware	Supported on: • u-blox 9 with protocol version 27
Type	Set
Comment	Overview: • This message is used to set a configuration by providing configuration data (a list of key and value pairs), which identify the configuration parameters to change, and their new values. • This message is limited to containing a maximum of 64 key-value pairs. • This message can be used multiple times and every time the result will be applied immediately. To send this message multiple times with the result being applied at the end, see version 1 of `UBX-CFG-VALSET` that supports transactions. • See Receiver Configuration for details. This message returns a UBX-ACK-NAK and no configuration is applied: • if any key is unknown to the receiver FW • if the layers bitfield does not specify a layer to save a value to • if the requested configuration is not valid. The validity of a configuration is checked only if the message requests to apply the configuration to the RAM configuration layer. Notes: • If a key is sent multiple times within the same message, then the value eventually being applied is the last sent.

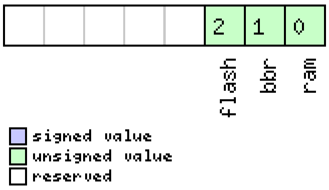
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x06	0x8A	4 + 1*N	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`version`	-	Message version, set to 0
1	X1	-	`layers`	-	The layers where the configuration should be applied (see graphic below)
2	U1[2]	-	`reserved1`	-	Reserved
Start of repeated block (N times)					
4 + 1*N	U1	-	`cfgData`	-	configuration data (key and value pairs)
End of repeated block					

Bitfield layers

This graphic explains the bits of `layers`



Name	*Description*
`ram`	Update configuration in the RAM layer
`bbr`	Update configuration in the BBR layer
`flash`	Update configuration in the Flash layer

6.9.31.2 Sets values corresponding to provided key-value pairs within a transaction

Message	**UBX-CFG-VALSET**
Description	**Sets values corresponding to provided key-value pairs within a transaction**
Firmware	Supported on: • u-blox 9 with protocol version 27
Type	Set
Comment	Overview: • This message is used to set a configuration by providing configuration data (a list of key and value pairs), which identify the configuration parameters to change, and their new values. • This message is limited to containing a maximum of 64 key-value pairs. • This message can be used multiple times with the result being managed within a transaction. Within a transaction there is no limit on the number key-value pairs; a transaction is effectively limited to the number of known keys. • See Receiver Configuration for details. • See version 0 of `UBX-CFG-VALSET` for simplified version of this message. This message returns a UBX-ACK-NAK, cancels any started transaction, and no configuration is applied: • if any key within a transaction is unknown to the receiver FW • if an invalid transaction state transition is requested • if the layers bitfield changes within a transaction • if the layers bitfield does not specify a layer to save a value to This message returns a UBX-ACK-NAK, and no configuration is applied: • if the requested configuration is not valid. While in a transaction context, only the last message that requests to apply the transaction returns a NAK. The validity of a configuration is checked only if the message requests to apply the configuration to the RAM configuration layer. This also applies to a transaction-less request. Notes: • Any request for another UBX-CFG-message type (including UBX-CFG-VALDEL and UBX-CFG-VALGET) will cancel any started transaction, and no configuration is applied. • This message can be sent with no key/values to set for the purposes of managing the transaction state transition. • If a key is sent multiple times within the same message or within the same transaction, then the value eventually being applied is the last sent.

	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*

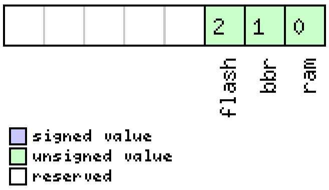
Message Structure	0xB5 0x62	0x06	0x8A	4 + 1*N	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`version`	-	Message version, set to 1
1	X1	-	`layers`	-	The layers where the configuration should be applied (see graphic below)
2	U1	-	`transaction`	-	Transaction action to be applied: (see graphic below)
3	U1	-	`reserved1`	-	Reserved
Start of repeated block (N times)					
4 + 1*N	U1	-	`cfgData`	-	configuration data (key and value pairs)
End of repeated block					

Bitfield layers

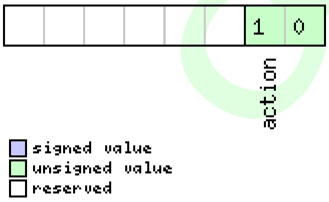
This graphic explains the bits of `layers`



Name	*Description*
`ram`	Update configuration in the RAM layer
`bbr`	Update configuration in the BBR layer
`flash`	Update configuration in the Flash layer

Bitfield transaction

This graphic explains the bits of `transaction`



Name	*Description*
`action`	Transaction action to be applied: 0: Transactionless UBX-CFG-VALSET: In the next UBX-CFG-VALSET, it can be either 0 or 1. If a transaction has not yet been started, the incoming configuration is applied (if valid). If a transaction has already been started, cancels any started transaction and the incoming configuration is applied (if valid). 1: (Re)Start set transaction: In the next UBX-CFG-VALSET, it can be either 0, 1, 2 or 3. If a transaction has not yet been started, a transaction will be started. If a transaction has already been started, restarts the transaction, effectively removing all previous non-applied UBX-CFG-VALSET messages. 2: Set transaction ongoing: In the next UBX-CFG-VALSET, it can be either 0, 1, 2 or 3. 3: Apply and end a set transaction: In the next UBX-CFG-VALSET, it can be either 0 or 1.

6.10 UBX-INF (0x04)

Information Messages: i.e. Printf-Style Messages, with IDs such as Error, Warning, Notice.

Messages in the INF class are used to output strings in a printf style from the firmware or application code. All INF messages have an associated type to indicate the kind of message.

6.10.1 UBX-INF-DEBUG (0x04 0x04)

6.10.1.1 ASCII output with debug contents

Message	**UBX-INF-DEBUG**					
Description	**ASCII output with debug contents**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Output					
Comment	This message has a variable length payload, representing an ASCII string.					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x04	0x04	0 + 1*N	*see below*	CK_A CK_B

Payload Contents:					
Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
Start of repeated block (N times)					
N*1	CH	-	`str`	-	ASCII Character
End of repeated block					

6.10.2 UBX-INF-ERROR (0x04 0x00)

6.10.2.1 ASCII output with error contents

Message	**UBX-INF-ERROR**					
Description	**ASCII output with error contents**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Output					
Comment	This message has a variable length payload, representing an ASCII string.					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x04	0x00	0 + 1*N	*see below*	CK_A CK_B

Payload Contents:					
Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
Start of repeated block (N times)					
N*1	CH	-	`str`	-	ASCII Character
End of repeated block					

6.10.3 UBX-INF-NOTICE (0x04 0x02)

6.10.3.1 ASCII output with informational contents

Message	**UBX-INF-NOTICE**
Description	**ASCII output with informational contents**
Firmware	Supported on: • u-blox 9 with protocol version 27
Type	Output
Comment	This message has a variable length payload, representing an ASCII string.

	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x04	0x02	0 + 1*N	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
Start of repeated block (N times)					
N*1	CH	-	`str`	-	ASCII Character
End of repeated block					

6.10.4 UBX-INF-TEST (0x04 0x03)

6.10.4.1 ASCII output with test contents

Message	**UBX-INF-TEST**
Description	**ASCII output with test contents**
Firmware	Supported on: • u-blox 9 with protocol version 27
Type	Output
Comment	This message has a variable length payload, representing an ASCII string.

	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x04	0x03	0 + 1*N	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
Start of repeated block (N times)					
N*1	CH	-	`str`	-	ASCII Character
End of repeated block					

6.10.5 UBX-INF-WARNING (0x04 0x01)

6.10.5.1 ASCII output with warning contents

Message	**UBX-INF-WARNING**					
Description	**ASCII output with warning contents**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Output					
Comment	This message has a variable length payload, representing an ASCII string.					
Message Structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xB5 0x62	0x04	0x01	0 + 1*N	*see below*	CK_A CK_B

Payload Contents:					
Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
Start of repeated block (N times)					
N*1	CH	-	`str`	-	ASCII Character
End of repeated block					

6.11 UBX-LOG (0x21)

Logging Messages: i.e. Log creation, deletion, info and retrieval.
Messages in the LOG class are used to configure and report status information of the logging and batching features.

6.11.1 UBX-LOG-CREATE (0x21 0x07)

6.11.1.1 Create Log File

Message	**UBX-LOG-CREATE**
Description	**Create Log File**
Firmware	Supported on: • u-blox 9 with protocol version 27
Type	Command
Comment	This message is used to create an initial logging file and activate the logging subsystem. `UBX-ACK-ACK` or `UBX-ACK-NAK` are returned to indicate success or failure. This message does not handle activation of recording or filtering of log entries (see `UBX-CFG-LOGFILTER`).

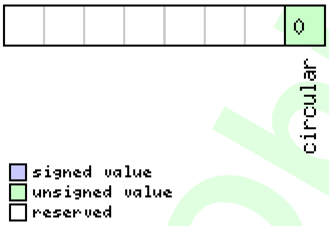
	Header	Class	ID	Length (Bytes)	Payload	Checksum
Message Structure	0xB5 0x62	0x21	0x07	8	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`version`	-	The version of this message. Set to 0
1	X1	-	`logCfg`	-	Config flags (see graphic below)
2	U1	-	`reserved1`	-	Reserved
3	U1	-	`logSize`	-	Indicates the size of the log: 0 (maximum safe size): Ensures that logging will not be interrupted and enough space will be left available for all other uses of the filestore 1 (minimum size): 2 (user defined): See 'userDefinedSize' below
4	U4	-	`userDefinedSize`	bytes	Sets the maximum amount of space in the filestore that can be used by the logging task. This field is only applicable if logSize is set to user defined.

Bitfield logCfg

This graphic explains the bits of `logCfg`



Name	Description
circular	Log is circular (new entries overwrite old ones in a full log) if this bit set

6.11.2 UBX-LOG-ERASE (0x21 0x03)

6.11.2.1 Erase Logged Data

Message	**UBX-LOG-ERASE**					
Description	**Erase Logged Data**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Command					
Comment	This message deactivates the logging system and erases all logged data. UBX-ACK-ACK or UBX-ACK-NAK are returned to indicate success or failure.					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x21	0x03	0	*see below*	CK_A CK_B
No payload						

6.11.3 UBX-LOG-FINDTIME (0x21 0x0E)

6.11.3.1 Find index of a log entry based on a given time

Message	**UBX-LOG-FINDTIME**					
Description	**Find index of a log entry based on a given time**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Input					
Comment	This message can be used for a time-based search of a log. It can find the index of the first log entry with time equal to the given time, otherwise the index of the most recent entry with time less than the given time. This index can then be used with the UBX-LOG-RETRIEVE message to provide time-based retrieval of log entries. Searching a log is effective for a given time later than the base date (January 1st, 2004). Searching a log for a given time earlier than the base date will result in an 'entry not found' response. Searching a log for a given time greater than the last recorded entry's time will return the index of the last recorded entry.					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x21	0x0E	12	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	version	-	Message version (=0 for this version)
1	U1	-	type	-	Message type, 0 for request
2	U1[2]	-	reserved1	-	Reserved
4	U2	-	year	-	Year (1-65635) of UTC time
6	U1	-	month	-	Month (1-12) of UTC time
7	U1	-	day	-	Day (1-31) of UTC time
8	U1	-	hour	-	Hour (0-23) of UTC time
9	U1	-	minute	-	Minute (0-59) of UTC time

LOG-FINDTIME continued

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
10	U1	-	`second`	-	Second (0-60) of UTC time
11	U1	-	`reserved2`	-	Reserved

6.11.3.2 Response to FINDTIME request.

Message	**UBX-LOG-FINDTIME**					
Description	**Response to FINDTIME request.**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Output					
Comment	-					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x21	0x0E	8	*see below*	CK_A CK_B
Payload Contents:						

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`version`	-	Message version (=1 for this version)
1	U1	-	`type`	-	Message type, 1 for response
2	U1[2]	-	`reserved1`	-	Reserved
4	U4	-	`entryNumber`	-	Index of the first log entry with time = given time, otherwise index of the most recent entry with time < given time. If 0xFFFFFFFF, no log entry found with time <= given time. The indexing of log entries is zero based.

6.11.4 UBX-LOG-INFO (0x21 0x08)

6.11.4.1 Poll for log information

Message	**UBX-LOG-INFO**					
Description	**Poll for log information**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Poll Request					
Comment	Upon sending of this message, the receiver returns UBX-LOG-INFO as defined below.					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x21	0x08	0	*see below*	CK_A CK_B
No payload						

6.11.4.2 Log information

Message	**UBX-LOG-INFO**
Description	**Log information**
Firmware	Supported on: • u-blox 9 with protocol version 27
Type	Output
Comment	This message is used to report information about the logging subsystem. Note: • The reported maximum log size will be smaller than that originally specified in LOG-CREATE due to logging and filestore implementation overheads. • Log entries are compressed in a variable length fashion, so it may be difficult to predict log space usage with any precision. • There may be times when the receiver does not have an accurate time (e.g. if the week number is not yet known), in which case some entries will not have a timestamp. This may result in the oldest/newest entry time values not taking account of these entries.

	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x21	0x08	48	*see below*	CK_A CK_B

Payload Contents:

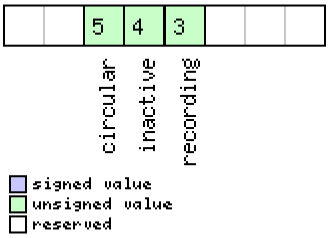
Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`version`	-	The version of this message. Set to 1
1	U1[3]	-	`reserved1`	-	Reserved
4	U4	-	`filestoreCapacity`	bytes	The capacity of the filestore
8	U1[8]	-	`reserved2`	-	Reserved
16	U4	-	`currentMaxLogSize`	bytes	The maximum size the current log is allowed to grow to
20	U4	-	`currentLogSize`	bytes	Approximate amount of space in log currently occupied
24	U4	-	`entryCount`	-	Number of entries in the log. Note: for circular logs this value will decrease when a group of entries is deleted to make space for new ones.
28	U2	-	`oldestYear`	-	Oldest entry UTC year (1-65635) or zero if there are no entries with known time
30	U1	-	`oldestMonth`	-	Oldest month (1-12)
31	U1	-	`oldestDay`	-	Oldest day (1-31)
32	U1	-	`oldestHour`	-	Oldest hour (0-23)
33	U1	-	`oldestMinute`	-	Oldest minute (0-59)
34	U1	-	`oldestSecond`	-	Oldest second (0-60)
35	U1	-	`reserved3`	-	Reserved
36	U2	-	`newestYear`	-	Newest year (1-65635) or zero if there are no entries with known time
38	U1	-	`newestMonth`	-	Newest month (1-12)
39	U1	-	`newestDay`	-	Newest day (1-31)
40	U1	-	`newestHour`	-	Newest hour (0-23)

LOG-INFO continued

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
41	U1	-	`newestMinute`	-	Newest minute (0-59)
42	U1	-	`newestSecond`	-	Newest second (0-60)
43	U1	-	`reserved4`	-	Reserved
44	X1	-	`status`	-	Log status flags (see graphic below)
45	U1[3]	-	`reserved5`	-	Reserved

Bitfield status

This graphic explains the bits of `status`



Name	*Description*
`recording`	Log entry recording is currently turned on
`inactive`	Logging system not active - no log present
`circular`	The current log is circular

6.11.5 UBX-LOG-RETRIEVEPOSEXTRA (0x21 0x0f)

6.11.5.1 Odometer log entry

Message	**UBX-LOG-RETRIEVEPOSEXTRA**					
Description	**Odometer log entry**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Output					
Comment	This message is used to report an odometer log entry					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x21	0x0f	32	*see below*	CK_A CK_B
Payload Contents:						

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U4	-	`entryIndex`	-	The index of this log entry
4	U1	-	`version`	-	The version of this message. Set to 0
5	U1	-	`reserved1`	-	Reserved
6	U2	-	`year`	-	Year (1-65635) of UTC time. Will be zero if time not known
8	U1	-	`month`	-	Month (1-12) of UTC time
9	U1	-	`day`	-	Day (1-31) of UTC time
10	U1	-	`hour`	-	Hour (0-23) of UTC time
11	U1	-	`minute`	-	Minute (0-59) of UTC time
12	U1	-	`second`	-	Second (0-60) of UTC time

LOG-RETRIEVEPOSEXTRA continued

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
13	U1[3]	-	reserved2	-	Reserved
16	U4	-	distance	-	Odometer distance traveled since the last time the odometer was reset by a UBX-NAV-RESETODO
20	U1[12]	-	reserved3	-	Reserved

6.11.6 UBX-LOG-RETRIEVEPOS (0x21 0x0b)

6.11.6.1 Position fix log entry

Message	**UBX-LOG-RETRIEVEPOS**				
Description	**Position fix log entry**				
Firmware	Supported on: • u-blox 9 with protocol version 27				
Type	Output				
Comment	This message is used to report a position fix log entry				
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x21	0x0b	40	*see below*	CK_A CK_B
Payload Contents:					

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U4	-	entryIndex	-	The index of this log entry
4	I4	1e-7	lon	deg	Longitude
8	I4	1e-7	lat	deg	Latitude
12	I4	-	hMSL	mm	Height above mean sea level
16	U4	-	hAcc	mm	Horizontal accuracy estimate
20	U4	-	gSpeed	mm/s	Ground speed (2-D)
24	U4	1e-5	heading	deg	Heading
28	U1	-	version	-	The version of this message. Set to 0
29	U1	-	fixType	-	Fix type: 0x01: Dead Reckoning only 0x02: 2D-Fix 0x03: 3D-Fix 0x04: GNSS + Dead Reckoning combined
30	U2	-	year	-	Year (1-65635) of UTC time
32	U1	-	month	-	Month (1-12) of UTC time
33	U1	-	day	-	Day (1-31) of UTC time
34	U1	-	hour	-	Hour (0-23) of UTC time
35	U1	-	minute	-	Minute (0-59) of UTC time
36	U1	-	second	-	Second (0-60) of UTC time
37	U1	-	reserved1	-	Reserved
38	U1	-	numSV	-	Number of satellites used in the position fix
39	U1	-	reserved2	-	Reserved

6.11.7 UBX-LOG-RETRIEVESTRING (0x21 0x0d)

6.11.7.1 Byte string log entry

Message	**UBX-LOG-RETRIEVESTRING**
Description	**Byte string log entry**
Firmware	Supported on: • u-blox 9 with protocol version 27
Type	Output
Comment	This message is used to report a byte string log entry

	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x21	0x0d	16 + 1*byteCount	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U4	-	`entryIndex`	-	The index of this log entry
4	U1	-	`version`	-	The version of this message. Set to 0
5	U1	-	`reserved1`	-	Reserved
6	U2	-	`year`	-	Year (1-65635) of UTC time. Will be zero if time not known
8	U1	-	`month`	-	Month (1-12) of UTC time
9	U1	-	`day`	-	Day (1-31) of UTC time
10	U1	-	`hour`	-	Hour (0-23) of UTC time
11	U1	-	`minute`	-	Minute (0-59) of UTC time
12	U1	-	`second`	-	Second (0-60) of UTC time
13	U1	-	`reserved2`	-	Reserved
14	U2	-	`byteCount`	-	Size of string in bytes
Start of repeated block (byteCount times)					
16 + 1*N	U1	-	`bytes`	-	The bytes of the string
End of repeated block					

6.11.8 UBX-LOG-RETRIEVE (0x21 0x09)

6.11.8.1 Request log data

Message	**UBX-LOG-RETRIEVE**
Description	**Request log data**
Firmware	Supported on: • u-blox 9 with protocol version 27
Type	Command
Comment	This message is used to request logged data (log recording must first be disabled, see `UBX-CFG-LOGFILTER`). Log entries are returned in chronological order, using the messages `UBX-LOG-RETRIEVEPOS` and `UBX-LOG-RETRIEVESTRING`. If the odometer was enabled at the time a position was logged, then message `UBX-LOG-RETRIEVEPOSEXTRA` will also be used. The maximum number of entries that can be returned in response to a single UBX-LOG-RETRIEVE message is 256. If more entries than this are required the message will need to be sent multiple times with different startNumbers. The retrieve will be stopped if any UBX-LOG message is received. The speed of transfer can be maximized by using a high

	data rate and temporarily stopping the GPS processing (see UBX-CFG-RST).					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x21	0x09	12	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U4	-	`startNumber`	-	Index of first log entry to be transferred. If it is larger than the index of the last available log entry, then the first log entry to be transferred is the last available log entry. The indexing of log entries is zero based.
4	U4	-	`entryCount`	-	Number of log entries to transfer in total including the first entry to be transferred. If it is larger than the log entries available starting from the first entry to be transferred, then only the available log entries are transferred followed by a UBX-ACK-NAK. The maximum is 256.
8	U1	-	`version`	-	The version of this message. Set to 0.
9	U1[3]	-	`reserved1`	-	Reserved

6.11.9 UBX-LOG-STRING (0x21 0x04)

6.11.9.1 Store arbitrary string in on-board flash

Message	**UBX-LOG-STRING**					
Description	**Store arbitrary string in on-board flash**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Command					
Comment	This message can be used to store an arbitrary byte string in the on-board flash memory. The maximum length that can be stored is 256 bytes.					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x21	0x04	0 + 1*N	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
Start of repeated block (N times)					
N*1	U1	-	`bytes`	-	The string of bytes to be logged (maximum 256)
End of repeated block					

6.12 UBX-MGA (0x13)

Multiple GNSS Assistance Messages: i.e. Assistance data for various GNSS.

Messages in the MGA class are used for GNSS aiding information from and to the receiver.

6.12.1 UBX-MGA-ACK (0x13 0x60)

6.12.1.1 UBX-MGA-ACK-DATA0

Message	**UBX-MGA-ACK-DATA0**
Description	**Multiple GNSS Acknowledge message**
Firmware	Supported on: • u-blox 9 with protocol version 27
Type	Output
Comment	This message is sent by a u-blox receiver to acknowledge the receipt of an assistance message. Acknowledgments are enabled by setting the ackAiding parameter in the `UBX-CFG-NAVX5` message. See the description of flow control for details.

	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x13	0x60	8	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`type`	-	Type of acknowledgment: 0: The message was not used by the receiver (see infoCode field for an indication of why) 1: The message was accepted for use by the receiver (the infoCode field will be 0)
1	U1	-	`version`	-	Message version (0x00 for this version)
2	U1	-	`infoCode`	-	Provides greater information on what the receiver chose to do with the message contents: 0: The receiver accepted the data 1: The receiver doesn't know the time so can't use the data (To resolve this a `UBX-MGA-INI-TIME_UTC` message should be supplied first) 2: The message version is not supported by the receiver 3: The message size does not match the message version 4: The message data could not be stored to the database 5: The receiver is not ready to use the message data 6: The message type is unknown
3	U1	-	`msgId`	-	UBX message ID of the ack'ed message
4	U1[4]	-	`msgPayloadStart`	-	The first 4 bytes of the ack'ed message's payload

6.12.2 UBX-MGA-BDS (0x13 0x03)

6.12.2.1 UBX-MGA-BDS-EPH

Message	**UBX-MGA-BDS-EPH**					
Description	**BDS Ephemeris Assistance**					
Firmware	Supported on: - u-blox 9 with protocol version 27					
Type	Input					
Comment	This message allows the delivery of BeiDou ephemeris assistance to a receiver. See the description of AssistNow Online for details.					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x13	0x03	88	*see below*	CK_A CK_B
Payload Contents:						

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`type`	-	Message type (0x01 for this type)
1	U1	-	`version`	-	Message version (0x00 for this version)
2	U1	-	`svId`	-	BDS satellite identifier (see Satellite Numbering)
3	U1	-	`reserved1`	-	Reserved
4	U1	-	`SatH1`	-	Autonomous satellite Health flag
5	U1	-	`IODC`	-	Issue of Data, Clock
6	I2	2^-66	`a2`	s/s^2	Time polynomial coefficient 2
8	I4	2^-50	`a1`	s/s	Time polynomial coefficient 1
12	I4	2^-33	`a0`	s	Time polynomial coefficient 0
16	U4	2^3	`toc`	s	Clock data reference time
20	I2	0.1	`TGD1`	ns	Equipment Group Delay Differential
22	U1	-	`URAI`	-	User Range Accuracy Index
23	U1	-	`IODE`	-	Issue of Data, Ephemeris
24	U4	2^3	`toe`	s	Ephemeris reference time
28	U4	2^-19	`sqrtA`	m^0.5	Square root of semi-major axis
32	U4	2^-33	`e`	-	Eccentricity
36	I4	2^-31	`omega`	semi-circles	Argument of perigee
40	I2	2^-43	`Deltan`	semi-circles/s	Mean motion difference from computed value
42	I2	2^-43	`IDOT`	semi-circles/s	Rate of inclination angle
44	I4	2^-31	`M0`	semi-circles	Mean anomaly at reference time
48	I4	2^-31	`Omega0`	semi-circles	Longitude of ascending node of orbital of plane computed according to reference time
52	I4	2^-43	`OmegaDot`	semi-circles/s	Rate of right ascension
56	I4	2^-31	`i0`	semi-circles	Inclination angle at reference time
60	I4	2^-31	`Cuc`	semi-circles	Amplitude of cosine harmonic correction term to the argument of latitude

MGA-BDS continued

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
64	I4	2^-31	`Cus`	semi-circles	Amplitude of sine harmonic correction term to the argument of latitude
68	I4	2^-6	`Crc`	m	Amplitude of cosine harmonic correction term to the orbit radius
72	I4	2^-6	`Crs`	m	Amplitude of sine harmonic correction term to the orbit radius
76	I4	2^-31	`Cic`	semi-circles	Amplitude of cosine harmonic correction term to the angle of inclination
80	I4	2^-31	`Cis`	semi-circles	Amplitude of sine harmonic correction term to the angle of inclination
84	U1[4]	-	`reserved2`	-	Reserved

6.12.2.2 UBX-MGA-BDS-ALM

Message	**UBX-MGA-BDS-ALM**					
Description	**BDS Almanac Assistance**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Input					
Comment	This message allows the delivery of BeiDou almanac assistance to a receiver. See the description of AssistNow Online for details.					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x13	0x03	40	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`type`	-	Message type (0x02 for this version)
1	U1	-	`version`	-	Message version (0x00 for this version)
2	U1	-	`svId`	-	BeiDou satellite identifier (see Satellite Numbering)
3	U1	-	`reserved1`	-	Reserved
4	U1	-	`Wna`	week	Almanac Week Number
5	U1	2^12	`toa`	s	Almanac reference time
6	I2	2^-19	`deltaI`	semi-circles	Almanac correction of orbit reference inclination at reference time
8	U4	2^-11	`sqrtA`	m^0.5	Almanac square root of semi-major axis
12	U4	2^-21	`e`	-	Almanac eccentricity
16	I4	2^-23	`omega`	semi-circles	Almanac argument of perigee
20	I4	2^-23	`M0`	semi-circles	Almanac mean anomaly at reference time
24	I4	2^-23	`Omega0`	semi-circles	Almanac longitude of ascending node of orbit plane at computed according to reference time

MGA-BDS continued

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
28	I4	2^-38	`omegaDot`	semi-circles/s	Almanac rate of right ascension
32	I2	2^-20	`a0`	s	Almanac satellite clock bias
34	I2	2^-38	`a1`	s/s	Almanac satellite clock rate
36	U1[4]	-	`reserved2`	-	Reserved

6.12.2.3 UBX-MGA-BDS-HEALTH

Message	**UBX-MGA-BDS-HEALTH**					
Description	**BDS Health Assistance**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Input					
Comment	This message allows the delivery of BeiDou health assistance to a receiver. See the description of AssistNow Online for details.					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x13	0x03	68	*see below*	CK_A CK_B
Payload Contents:						

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`type`	-	Message type (0x04 for this type)
1	U1	-	`version`	-	Message version (0x00 for this version)
2	U1[2]	-	`reserved1`	-	Reserved
4	U2[30]	-	`healthCode`	-	Each two-byte value represents a BDS SV (1-30). The 9 LSBs of each byte contain the 9 bit health code from subframe 5 pages 7,8 of the D1 message, and from subframe 5 pages 35,36 of the D1 message.
64	U1[4]	-	`reserved2`	-	Reserved

6.12.2.4 UBX-MGA-BDS-UTC

Message	**UBX-MGA-BDS-UTC**					
Description	**BDS UTC Assistance**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Input					
Comment	This message allows the delivery of BeiDou UTC assistance to a receiver. See the description of AssistNow Online for details.					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x13	0x03	20	*see below*	CK_A CK_B
Payload Contents:						

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`type`	-	Message type (0x05 for this type)

MGA-BDS continued

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
1	U1	-	`version`	-	Message version (0x00 for this version)
2	U1[2]	-	`reserved1`	-	Reserved
4	I4	2^-30	`a0UTC`	s	BDT clock bias relative to UTC
8	I4	2^-50	`a1UTC`	s/s	BDT clock rate relative to UTC
12	I1	-	`dtLS`	s	Delta time due to leap seconds before the new leap second effective
13	U1[1]	-	`reserved2`	-	Reserved
14	U1	-	`wnRec`	week	BeiDou week number of reception of this UTC parameter set (8 bit truncated)
15	U1	-	`wnLSF`	week	Week number of the new leap second
16	U1	-	`dN`	day	Day number of the new leap second
17	I1	-	`dtLSF`	s	Delta time due to leap seconds after the new leap second effective
18	U1[2]	-	`reserved3`	-	Reserved

6.12.2.5 UBX-MGA-BDS-IONO

Message	**UBX-MGA-BDS-IONO**					
Description	**BDS Ionospheric Assistance**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Input					
Comment	This message allows the delivery of BeiDou ionospheric assistance to a receiver. See the description of AssistNow Online for details.					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x13	0x03	16	*see below*	CK_A CK_B
Payload Contents:						

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`type`	-	Message type (0x06 for this type)
1	U1	-	`version`	-	Message version (0x00 for this version)
2	U1[2]	-	`reserved1`	-	Reserved
4	I1	2^-30	`alpha0`	s	Ionospheric parameter alpha0
5	I1	2^-27	`alpha1`	s/pi	Ionospheric parameter alpha1
6	I1	2^-24	`alpha2`	s/pi^2	Ionospheric parameter alpha2
7	I1	2^-24	`alpha3`	s/pi^3	Ionospheric parameter alpha3
8	I1	2^11	`beta0`	s	Ionospheric parameter beta0
9	I1	2^14	`beta1`	s/pi	Ionospheric parameter beta1
10	I1	2^16	`beta2`	s/pi^2	Ionospheric parameter beta2
11	I1	2^16	`beta3`	s/pi^3	Ionospheric parameter beta3
12	U1[4]	-	`reserved2`	-	Reserved

6.12.3 UBX-MGA-DBD (0x13 0x80)

6.12.3.1 Poll the Navigation Database

Message	**UBX-MGA-DBD**
Description	**Poll the Navigation Database**
Firmware	Supported on: • u-blox 9 with protocol version 27
Type	Poll Request
Comment	Poll the whole navigation data base. The receiver will send all available data from its internal database. The receiver will indicate the finish of the transmission with a UBX-MGA-ACK. The msgPayloadStart field of the UBX-MGA-ACK message will contain a U4 representing the number of UBX-MGA-DBD-DATA* messages sent.

	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x13	0x80	0	*see below*	CK_A CK_B
No payload						

6.12.3.2 Navigation Database Dump Entry

Message	**UBX-MGA-DBD**
Description	**Navigation Database Dump Entry**
Firmware	Supported on: • u-blox 9 with protocol version 27
Type	Input/Output
Comment	**UBX-MGA-DBD messages are only intended to be sent back to the same receiver that generated them.** Navigation database entry. The data fields are firmware specific. Transmission of this type of message will be acknowledged by UBX-MGA-ACK messages, if acknowledgment has been enabled (see the description of flow control for details). The maximum payload size for firmware 2.01 onwards is 164 bytes (which makes the maximum message size 172 bytes).

	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x13	0x80	12 + 1*N	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1[12]	-	`reserved1`	-	Reserved
Start of repeated block (N times)					
12 + 1*N	U1	-	`data`	-	fw specific data
End of repeated block					

6.12.4 UBX-MGA-GAL (0x13 0x02)

6.12.4.1 UBX-MGA-GAL-EPH

Message	**UBX-MGA-GAL-EPH**				
Description	**Galileo Ephemeris Assistance**				
Firmware	Supported on: - u-blox 9 with protocol version 27				
Type	Input				
Comment	This message allows the delivery of Galileo ephemeris assistance to a receiver. See the description of AssistNow Online for details.				
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x13	0x02	76	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`type`	-	Message type (0x01 for this type)
1	U1	-	`version`	-	Message version (0x00 for this version)
2	U1	-	`svId`	-	Galileo Satellite identifier (see Satellite Numbering)
3	U1	-	`reserved1`	-	Reserved
4	U2	-	`iodNav`	-	Ephemeris and clock correction Issue of Data
6	I2	2^-43	`deltaN`	semi-circles/s	Mean motion difference from computed value
8	I4	2^-31	`m0`	semi-circles	Mean anomaly at reference time
12	U4	2^-33	`e`	-	Eccentricity
16	U4	2^-19	`sqrtA`	m^0.5	Square root of the semi-major axis
20	I4	2^-31	`omega0`	semi-circles	Longitude of ascending node of orbital plane at weekly epoch
24	I4	2^-31	`i0`	semi-circles	Inclination angle at reference time
28	I4	2^-31	`omega`	semi-circles	Argument of perigee
32	I4	2^-43	`omegaDot`	semi-circles/s	Rate of change of right ascension
36	I2	2^-43	`iDot`	semi-circles/s	Rate of change of inclination angle
38	I2	2^-29	`cuc`	radians	Amplitude of the cosine harmonic correction term to the argument of latitude
40	I2	2^-29	`cus`	radians	Amplitude of the sine harmonic correction term to the argument of latitude
42	I2	2^-5	`crc`	radians	Amplitude of the cosine harmonic correction term to the orbit radius
44	I2	2^-5	`crs`	radians	Amplitude of the sine harmonic correction term to the orbit radius
46	I2	2^-29	`cic`	radians	Amplitude of the cosine harmonic correction term to the angle of inclination

MGA-GAL continued

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
48	I2	2^-29	`cis`	radians	Amplitude of the sine harmonic correction term to the angle of inclination
50	U2	60	`toe`	s	Ephemeris reference time
52	I4	2^-34	`af0`	s	SV clock bias correction coefficient
56	I4	2^-46	`af1`	s/s	SV clock drift correction coefficient
60	I1	2^-59	`af2`	s/s squared	SV clock drift rate correction coefficient
61	U1	-	`sisaIndexE1E5b`	-	Signal-In-Space Accuracy index for dual frequency E1-E5b
62	U2	60	`toc`	s	Clock correction data reference Time of Week
64	I2	-	`bgdE1E5b`	-	E1-E5b Broadcast Group Delay
66	U1[2]	-	`reserved2`	-	Reserved
68	U1	-	`healthE1B`	-	E1-B Signal Health Status
69	U1	-	`dataValidityE1B`	-	E1-B Data Validity Status
70	U1	-	`healthE5b`	-	E5b Signal Health Status
71	U1	-	`dataValidityE5b`	-	E5b Data Validity Status
72	U1[4]	-	`reserved3`	-	Reserved

6.12.4.2 UBX-MGA-GAL-ALM

Message	**UBX-MGA-GAL-ALM**					
Description	**Galileo Almanac Assistance**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Input					
Comment	This message allows the delivery of Galileo almanac assistance to a receiver. See the description of AssistNow Online for details.					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x13	0x02	32	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`type`	-	Message type (0x02 for this type)
1	U1	-	`version`	-	Message version (0x00 for this version)
2	U1	-	`svId`	-	Galileo Satellite identifier (see Satellite Numbering)
3	U1	-	`reserved1`	-	Reserved
4	U1	-	`ioda`	-	Almanac Issue of Data
5	U1	-	`almWNa`	week	Almanac reference week number
6	U2	600	`toa`	s	Almanac reference time
8	I2	2^-9	`deltaSqrtA`	m^0.5	Difference with respect to the square root of the nominal semi-major axis (29 600 km)

MGA-GAL continued

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
10	U2	2^-16	`e`	-	Eccentricity
12	I2	2^-14	`deltaI`	semi-circles	Inclination at reference time relative to i0 = 56 degree
14	I2	2^-15	`omega0`	semi-circles	Longitude of ascending node of orbital plane at weekly epoch
16	I2	2^-33	`omegaDot`	semi-circles/s	Rate of change of right ascension
18	I2	2^-15	`omega`	semi-circles	Argument of perigee
20	I2	2^-15	`m0`	semi-circles	Satellite mean anomaly at reference time
22	I2	2^-19	`af0`	s	Satellite clock correction bias 'truncated'
24	I2	2^-38	`af1`	s/s	Satellite clock correction linear 'truncated'
26	U1	-	`healthE1B`	-	Satellite E1-B signal health status
27	U1	-	`healthE5b`	-	Satellite E5b signal health status
28	U1[4]	-	`reserved2`	-	Reserved

6.12.4.3 UBX-MGA-GAL-TIMEOFFSET

Message	**UBX-MGA-GAL-TIMEOFFSET**					
Description	**Galileo GPS time offset assistance**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Input					
Comment	This message allows the delivery of Galileo time to GPS time offset. See the description of AssistNow Online for details.					
Message Structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xB5 0x62	0x13	0x02	12	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`type`	-	Message type (0x03 for this type)
1	U1	-	`version`	-	Message version (0x00 for this version)
2	U1[2]	-	`reserved1`	-	Reserved
4	I2	2^-35	`a0G`	s	Constant term of the polynomial describing the offset
6	I2	2^-51	`a1G`	s/s	Rate of change of the offset
8	U1	3600	`t0G`	s	DReference time for GGTO data
9	U1	-	`wn0G`	weeks	Week Number of GGTO reference
10	U1[2]	-	`reserved2`	-	Reserved

6.12.4.4 UBX-MGA-GAL-UTC

Message	**UBX-MGA-GAL-UTC**					
Description	**Galileo UTC Assistance**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Input					
Comment	This message allows the delivery of Galileo UTC assistance to a receiver. See the description of AssistNow Online for details.					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x13	0x02	20	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`type`	-	Message type (0x05 for this type)
1	U1	-	`version`	-	Message version (0x00 for this version)
2	U1[2]	-	`reserved1`	-	Reserved
4	I4	2^-30	`a0`	s	First parameter of UTC polynomial
8	I4	2^-50	`a1`	s/s	Second parameter of UTC polynomial
12	I1	-	`dtLS`	s	Delta time due to current leap seconds
13	U1	3600	`tot`	s	UTC parameters reference time of week (Galileo time)
14	U1	-	`wnt`	weeks	UTC parameters reference week number (the 8 bit WNt field)
15	U1	-	`wnLSF`	weeks	Week number at the end of which the future leap second becomes effective (the 8 bit WNLSF field)
16	U1	-	`dN`	days	Day number at the end of which the future leap second becomes effective
17	I1	-	`dTLSF`	s	Delta time due to future leap seconds
18	U1[2]	-	`reserved2`	-	Reserved

6.12.5 UBX-MGA-GLO (0x13 0x06)

6.12.5.1 UBX-MGA-GLO-EPH

Message	**UBX-MGA-GLO-EPH**					
Description	**GLONASS Ephemeris Assistance**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Input					
Comment	This message allows the delivery of GLONASS ephemeris assistance to a receiver. See the description of AssistNow Online for details.					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x13	0x06	48	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*

MGA-GLO continued

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`type`	-	Message type (0x01 for this type)
1	U1	-	`version`	-	Message version (0x00 for this version)
2	U1	-	`svId`	-	GLONASS Satellite identifier (see Satellite Numbering)
3	U1	-	`reserved1`	-	Reserved
4	U1	-	`FT`	-	User range accuracy
5	U1	-	`B`	-	Health flag from string 2
6	U1	-	`M`	-	Type of GLONASS satellite (1 indicates GLONASS-M)
7	I1	-	`H`	-	Carrier frequency number of navigation RF signal, Range=(-7 .. 6), -128 for unknown
8	I4	2^-11	`x`	km	X component of the SV position in PZ-90.02 coordinate System
12	I4	2^-11	`y`	km	Y component of the SV position in PZ-90.02 coordinate System
16	I4	2^-11	`z`	km	Z component of the SV position in PZ-90.02 coordinate System
20	I4	2^-20	`dx`	km/s	X component of the SV velocity in PZ-90.02 coordinate System
24	I4	2^-20	`dy`	km/s	Y component of the SV velocity in PZ-90.02 coordinate System
28	I4	2^-20	`dz`	km/s	Z component of the SV velocity in PZ-90.02 coordinate System
32	I1	2^-30	`ddx`	km/s^2	X component of the SV acceleration in PZ-90.02 coordinate System
33	I1	2^-30	`ddy`	km/s^2	Y component of the SV acceleration in PZ-90.02 coordinate System
34	I1	2^-30	`ddz`	km/s^2	Z component of the SV acceleration in PZ-90.02 coordinate System
35	U1	15	`tb`	minutes	Index of a time interval within current day according to UTC(SU)
36	I2	2^-40	`gamma`	-	Relative carrier frequency deviation
38	U1	-	`E`	days	Ephemeris data age indicator
39	I1	2^-30	`deltaTau`	s	Time difference between L2 and L1 band
40	I4	2^-30	`tau`	s	SV clock bias
44	U1[4]	-	`reserved2`	-	Reserved

6.12.5.2 UBX-MGA-GLO-ALM

Message	**UBX-MGA-GLO-ALM**					
Description	**GLONASS Almanac Assistance**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Input					
Comment	This message allows the delivery of GLONASS almanac assistance to a receiver. See the description of AssistNow Online for details.					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x13	0x06	36	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`type`	-	Message type (0x02 for this type)
1	U1	-	`version`	-	Message version (0x00 for this version)
2	U1	-	`svId`	-	GLONASS Satellite identifier (see Satellite Numbering)
3	U1	-	`reserved1`	-	Reserved
4	U2	-	`N`	days	Reference calender day number of almanac within the four-year period (from string 5)
6	U1	-	`M`	-	Type of GLONASS satellite (1 indicates GLONASS-M)
7	U1	-	`C`	-	Unhealthy flag at instant of almanac upload (1 indicates operability of satellite)
8	I2	2^-18	`tau`	s	Coarse time correction to GLONASS time
10	U2	2^-20	`epsilon`	-	Eccentricity
12	I4	2^-20	`lambda`	semi-circles	Longitude of the first (within the N-day) ascending node of satellite orbit in PC-90.02 coordinate system
16	I4	2^-20	`deltaI`	semi-circles	Correction to the mean value of inclination
20	U4	2^-5	`tLambda`	s	Time of the first ascending node passage
24	I4	2^-9	`deltaT`	s/orbital-period	Correction to the mean value of Draconian period
28	I1	2^-14	`deltaDT`	s/orbital-period^2	Rate of change of Draconian period
29	I1	-	`H`	-	Carrier frequency number of navigation RF signal, Range=(-7 .. 6)
30	I2	-	`omega`	-	Argument of perigee
32	U1[4]	-	`reserved2`	-	Reserved

6.12.5.3 UBX-MGA-GLO-TIMEOFFSET

Message	**UBX-MGA-GLO-TIMEOFFSET**
Description	**GLONASS Auxiliary Time Offset Assistance**
Firmware	Supported on: • u-blox 9 with protocol version 27
Type	Input
Comment	This message allows the delivery of auxiliary GLONASS assistance (including the GLONASS time offsets to other GNSS systems) to a receiver. See the description of AssistNow Online for details.

	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x13	0x06	20	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`type`	-	Message type (0x03 for this type)
1	U1	-	`version`	-	Message version (0x00 for this version)
2	U2	-	`N`	days	Reference calendar day number within the four-year period of almanac (from string 5)
4	I4	2^-27	`tauC`	s	Time scale correction to UTC(SU) time
8	I4	2^-31	`tauGps`	s	Correction to GPS time relative to GLONASS time
12	I2	2^-10	`B1`	s	Coefficient to determine delta UT1
14	I2	2^-16	`B2`	s/msd	Rate of change of delta UT1
16	U1[4]	-	`reserved1`	-	Reserved

6.12.6 UBX-MGA-GPS (0x13 0x00)

6.12.6.1 UBX-MGA-GPS-EPH

Message	**UBX-MGA-GPS-EPH**
Description	**GPS Ephemeris Assistance**
Firmware	Supported on: • u-blox 9 with protocol version 27
Type	Input
Comment	This message allows the delivery of GPS ephemeris assistance to a receiver. See the description of AssistNow Online for details.

	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x13	0x00	68	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`type`	-	Message type (0x01 for this type)
1	U1	-	`version`	-	Message version (0x00 for this version)
2	U1	-	`svId`	-	GPS Satellite identifier (see Satellite Numbering)
3	U1	-	`reserved1`	-	Reserved
4	U1	-	`fitInterval`	-	Fit interval flag
5	U1	-	`uraIndex`	-	URA index

MGA-GPS continued

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
6	U1	-	svHealth	-	SV health
7	I1	2^-31	tgd	s	Group delay differential
8	U2	-	iodc	-	IODC
10	U2	2^4	toc	s	Clock data reference time
12	U1	-	reserved2	-	Reserved
13	I1	2^-55	af2	s/s squared	Time polynomial coefficient 2
14	I2	2^-43	af1	s/s	Time polynomial coefficient 1
16	I4	2^-31	af0	s	Time polynomial coefficient 0
20	I2	2^-5	crs	m	Crs
22	I2	2^-43	deltaN	semi-circles/s	Mean motion difference from computed value
24	I4	2^-31	m0	semi-circles	Mean anomaly at reference time
28	I2	2^-29	cuc	radians	Amplitude of cosine harmonic correction term to argument of latitude
30	I2	2^-29	cus	radians	Amplitude of sine harmonic correction term to argument of latitude
32	U4	2^-33	e	-	Eccentricity
36	U4	2^-19	sqrtA	m^0.5	Square root of the semi-major axis
40	U2	2^4	toe	s	Reference time of ephemeris
42	I2	2^-29	cic	radians	Amplitude of cos harmonic correction term to angle of inclination
44	I4	2^-31	omega0	semi-circles	Longitude of ascending node of orbit plane at weekly epoch
48	I2	2^-29	cis	radians	Amplitude of sine harmonic correction term to angle of inclination
50	I2	2^-5	crc	m	Amplitude of cosine harmonic correction term to orbit radius
52	I4	2^-31	i0	semi-circles	Inclination angle at reference time
56	I4	2^-31	omega	semi-circles	Argument of perigee
60	I4	2^-43	omegaDot	semi-circles/s	Rate of right ascension
64	I2	2^-43	idot	semi-circles/s	Rate of inclination angle
66	U1[2]	-	reserved3	-	Reserved

6.12.6.2 UBX-MGA-GPS-ALM

Message	**UBX-MGA-GPS-ALM**						
Description	**GPS Almanac Assistance**						
Firmware	Supported on: • u-blox 9 with protocol version 27						
Type	Input						
Comment	This message allows the delivery of GPS almanac assistance to a receiver. See the description of AssistNow Online for details.						
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*	
Message Structure	0xB5 0x62	0x13	0x00	36	*see below*	CK_A CK_B	

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`type`	-	Message type (0x02 for this type)
1	U1	-	`version`	-	Message version (0x00 for this version)
2	U1	-	`svId`	-	GPS Satellite identifier (see Satellite Numbering)
3	U1	-	`svHealth`	-	SV health information
4	U2	2^-21	`e`	-	Eccentricity
6	U1	-	`almWNa`	week	Reference week number of almanac (the 8 bit WNa field)
7	U1	2^12	`toa`	s	Reference time of almanac
8	I2	2^-19	`deltaI`	semi-circles	Delta inclination angle at reference time
10	I2	2^-38	`omegaDot`	semi-circles/s	Rate of right ascension
12	U4	2^-11	`sqrtA`	m^0.5	Square root of the semi-major axis
16	I4	2^-23	`omega0`	semi-circles	Longitude of ascending node of orbit plane
20	I4	2^-23	`omega`	semi-circles	Argument of perigee
24	I4	2^-23	`m0`	semi-circles	Mean anomaly at reference time
28	I2	2^-20	`af0`	s	Time polynomial coefficient 0 (8 MSBs)
30	I2	2^-38	`af1`	s/s	Time polynomial coefficient 1
32	U1[4]	-	`reserved1`	-	Reserved

6.12.6.3 UBX-MGA-GPS-HEALTH

Message	**UBX-MGA-GPS-HEALTH**
Description	**GPS Health Assistance**
Firmware	Supported on: • u-blox 9 with protocol version 27
Type	Input
Comment	This message allows the delivery of GPS health assistance to a receiver. See the description of AssistNow Online for details.

	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x13	0x00	40	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`type`	-	Message type (0x04 for this type)
1	U1	-	`version`	-	Message version (0x00 for this version)
2	U1[2]	-	`reserved1`	-	Reserved
4	U1[32]	-	`healthCode`	-	Each byte represents a GPS SV (1-32). The 6 LSBs of each byte contains the 6 bit health code from subframes 4/5 page 25.
36	U1[4]	-	`reserved2`	-	Reserved

6.12.6.4 UBX-MGA-GPS-UTC

Message	**UBX-MGA-GPS-UTC**
Description	**GPS UTC Assistance**
Firmware	Supported on: • u-blox 9 with protocol version 27
Type	Input
Comment	This message allows the delivery of GPS UTC assistance to a receiver. See the description of AssistNow Online for details.

	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x13	0x00	20	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`type`	-	Message type (0x05 for this type)
1	U1	-	`version`	-	Message version (0x00 for this version)
2	U1[2]	-	`reserved1`	-	Reserved
4	I4	2^-30	`utcA0`	s	First parameter of UTC polynomial
8	I4	2^-50	`utcA1`	s/s	Second parameter of UTC polynomial
12	I1	-	`utcDtLS`	s	Delta time due to current leap seconds
13	U1	2^12	`utcTot`	s	UTC parameters reference time of week (GPS time)
14	U1	-	`utcWNt`	weeks	UTC parameters reference week number (the 8 bit WNt field)

MGA-GPS continued

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
15	U1	-	`utcWNlsf`	weeks	Week number at the end of which the future leap second becomes effective (the 8 bit WNLSF field)
16	U1	-	`utcDn`	days	Day number at the end of which the future leap second becomes effective
17	I1	-	`utcDtLSF`	s	Delta time due to future leap seconds
18	U1[2]	-	`reserved2`	-	Reserved

6.12.6.5 UBX-MGA-GPS-IONO

Message	**UBX-MGA-GPS-IONO**				
Description	**GPS Ionosphere Assistance**				
Firmware	Supported on: • u-blox 9 with protocol version 27				
Type	Input				
Comment	This message allows the delivery of GPS ionospheric assistance to a receiver. See the description of AssistNow Online for details.				
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x13	0x00	16	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`type`	-	Message type (0x06 for this type)
1	U1	-	`version`	-	Message version (0x00 for this version)
2	U1[2]	-	`reserved1`	-	Reserved
4	I1	2^-30	`ionoAlpha0`	s	Ionospheric parameter alpha0 [s]
5	I1	2^-27	`ionoAlpha1`	s/semi-circle	Ionospheric parameter alpha1 [s/semi-circle]
6	I1	2^-24	`ionoAlpha2`	s/(semi-circle^2)	Ionospheric parameter alpha2 [s/semi-circle^2]
7	I1	2^-24	`ionoAlpha3`	s/(semi-circle^3)	Ionospheric parameter alpha3 [s/semi-circle^3]
8	I1	2^11	`ionoBeta0`	s	Ionospheric parameter beta0 [s]
9	I1	2^14	`ionoBeta1`	s/semi-circle	Ionospheric parameter beta1 [s/semi-circle]
10	I1	2^16	`ionoBeta2`	s/(semi-circle^2)	Ionospheric parameter beta2 [s/semi-circle^2]
11	I1	2^16	`ionoBeta3`	s/(semi-circle^3)	Ionospheric parameter beta3 [s/semi-circle^3]
12	U1[4]	-	`reserved2`	-	Reserved

6.12.7 UBX-MGA-INI (0x13 0x40)

6.12.7.1 UBX-MGA-INI-POS_XYZ

Message	**UBX-MGA-INI-POS_XYZ**				
Description	**Initial Position Assistance**				
Firmware	Supported on: - u-blox 9 with protocol version 27				
Type	Input				
Comment	**Supplying position assistance that is inaccurate by more than the specified position accuracy, may lead to substantially degraded receiver performance.** This message allows the delivery of initial position assistance to a receiver in cartesian ECEF coordinates. This message is equivalent to the UBX-MGA-INI-POS_LLH message, except for the coordinate system. See the description of AssistNow Online for details.				
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x13	0x40	20	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`type`	-	Message type (0x00 for this type)
1	U1	-	`version`	-	Message version (0x00 for this version)
2	U1[2]	-	`reserved1`	-	Reserved
4	I4	-	`ecefX`	cm	WGS84 ECEF X coordinate
8	I4	-	`ecefY`	cm	WGS84 ECEF Y coordinate
12	I4	-	`ecefZ`	cm	WGS84 ECEF Z coordinate
16	U4	-	`posAcc`	cm	Position accuracy (stddev)

6.12.7.2 UBX-MGA-INI-POS_LLH

Message	**UBX-MGA-INI-POS_LLH**					
Description	**Initial Position Assistance**					
Firmware	Supported on: - u-blox 9 with protocol version 27					
Type	Input					
Comment	**Supplying position assistance that is inaccurate by more than the specified position accuracy, may lead to substantially degraded receiver performance.** This message allows the delivery of initial position assistance to a receiver in WGS84 lat/long/alt coordinates. This message is equivalent to the UBX-MGA-INI-POS_XYZ message, except for the coordinate system. See the description of AssistNow Online for details.					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x13	0x40	20	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`type`	-	Message type (0x01 for this type)
1	U1	-	`version`	-	Message version (0x00 for this version)
2	U1[2]	-	`reserved1`	-	Reserved

MGA-INI continued

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
4	I4	1e-7	`lat`	deg	WGS84 Latitude
8	I4	1e-7	`lon`	deg	WGS84 Longitude
12	I4	-	`alt`	cm	WGS84 Altitude
16	U4	-	`posAcc`	cm	Position accuracy (stddev)

6.12.7.3 UBX-MGA-INI-TIME_UTC

Message	**UBX-MGA-INI-TIME_UTC**
Description	**Initial Time Assistance**
Firmware	Supported on: • u-blox 9 with protocol version 27
Type	Input
Comment	**Supplying time assistance that is inaccurate by more than the specified time accuracy, may lead to substantially degraded receiver performance.** This message allows the delivery of UTC time assistance to a receiver. This message is equivalent to the `UBX-MGA-INI-TIME_GNSS` message, except for the time base. See the description of AssistNow Online for details.

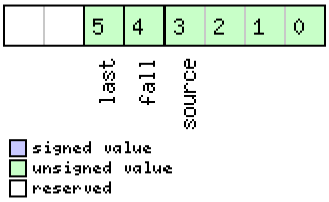
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x13	0x40	24	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`type`	-	Message type (0x10 for this type)
1	U1	-	`version`	-	Message version (0x00 for this version)
2	X1	-	`ref`	-	Reference to be used to set time (see graphic below)
3	I1	-	`leapSecs`	s	Number of leap seconds since 1980 (or 0x80 = -128 if unknown)
4	U2	-	`year`	-	Year
6	U1	-	`month`	-	Month, starting at 1
7	U1	-	`day`	-	Day, starting at 1
8	U1	-	`hour`	-	Hour, from 0 to 23
9	U1	-	`minute`	-	Minute, from 0 to 59
10	U1	-	`second`	s	Seconds, from 0 to 59
11	U1	-	`reserved1`	-	Reserved
12	U4	-	`ns`	ns	Nanoseconds, from 0 to 999,999,999
16	U2	-	`tAccS`	s	Seconds part of time accuracy
18	U1[2]	-	`reserved2`	-	Reserved
20	U4	-	`tAccNs`	ns	Nanoseconds part of time accuracy, from 0 to 999,999,999

Bitfield ref

This graphic explains the bits of `ref`



Name	Description
`source`	0: none, i.e. on receipt of message (will be inaccurate!) 1: relative to pulse sent to EXTINT0 2: relative to pulse sent to EXTINT1 3-15: reserved
`fall`	use falling edge of EXTINT pulse (default rising) - only if source is EXTINT
`last`	use last EXTINT pulse (default next pulse) - only if source is EXTINT

6.12.7.4 UBX-MGA-INI-TIME_GNSS

Message	**UBX-MGA-INI-TIME_GNSS**					
Description	**Initial Time Assistance**					
Firmware	Supported on: - u-blox 9 with protocol version 27					
Type	Input					
Comment	**Supplying time assistance that is inaccurate by more than the specified time accuracy, may lead to substantially degraded receiver performance.** This message allows the delivery of time assistance to a receiver in a chosen GNSS timebase. This message is equivalent to the `UBX-MGA-INI-TIME_UTC` message, except for the time base. See the description of AssistNow Online for details.					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x13	0x40	24	*see below*	CK_A CK_B

Payload Contents:

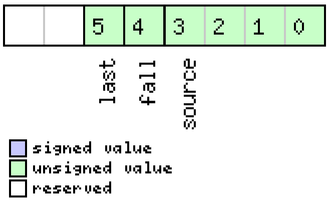
Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`type`	-	Message type (0x11 for this type)
1	U1	-	`version`	-	Message version (0x00 for this version)
2	X1	-	`ref`	-	Reference to be used to set time (see graphic below)
3	U1	-	`gnssId`	-	Source of time information. Currently supported: 0: GPS time 2: Galileo time 3: BeiDou time 6: GLONASS time: week = 834 + ((N4-1)*1461 + Nt)/7, tow = (((N4-1)*1461 + Nt) % 7) * 86400 + tod
4	U1[2]	-	`reserved1`	-	Reserved
6	U2	-	`week`	-	GNSS week number
8	U4	-	`tow`	s	GNSS time of week

MGA-INI continued

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
12	U4	-	`ns`	ns	GNSS time of week, nanosecond part from 0 to 999,999,999
16	U2	-	`tAccS`	s	Seconds part of time accuracy
18	U1[2]	-	`reserved2`	-	Reserved
20	U4	-	`tAccNs`	ns	Nanoseconds part of time accuracy, from 0 to 999,999,999

Bitfield ref

This graphic explains the bits of `ref`



Name	*Description*
`source`	0: none, i.e. on receipt of message (will be inaccurate!) 1: relative to pulse sent to EXTINT0 2: relative to pulse sent to EXTINT1 3-15: reserved
`fall`	use falling edge of EXTINT pulse (default rising) - only if source is EXTINT
`last`	use last EXTINT pulse (default next pulse) - only if source is EXTINT

6.12.7.5 UBX-MGA-INI-CLKD

Message	**UBX-MGA-INI-CLKD**					
Description	**Initial Clock Drift Assistance**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Input					
Comment	**Supplying clock drift assistance that is inaccurate by more than the specified accuracy, may lead to substantially degraded receiver performance.** This message allows the delivery of clock drift assistance to a receiver. See the description of AssistNow Online for details.					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x13	0x40	12	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`type`	-	Message type (0x20 for this type)
1	U1	-	`version`	-	Message version (0x00 for this version)
2	U1[2]	-	`reserved1`	-	Reserved
4	I4	-	`clkD`	ns/s	Clock drift
8	U4	-	`clkDAcc`	ns/s	Clock drift accuracy

6.12.7.6 UBX-MGA-INI-FREQ

Message	**UBX-MGA-INI-FREQ**
Description	**Initial Frequency Assistance**
Firmware	Supported on: • u-blox 9 with protocol version 27
Type	Input
Comment	**Supplying external frequency assistance that is inaccurate by more than the specified accuracy, may lead to substantially degraded receiver performance.** This message allows the delivery of external frequency assistance to a receiver. See the description of AssistNow Online for details.

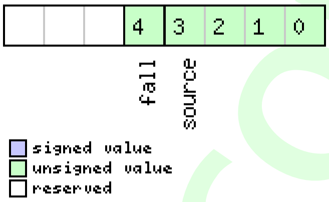
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x13	0x40	12	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`type`	-	Message type (0x21 for this type)
1	U1	-	`version`	-	Message version (0x00 for this version)
2	U1	-	`reserved1`	-	Reserved
3	X1	-	`flags`	-	Frequency reference (see graphic below)
4	I4	1e-2	`freq`	Hz	Frequency
8	U4	-	`freqAcc`	ppb	Frequency accuracy

Bitfield flags

This graphic explains the bits of `flags`



Name	*Description*
`source`	0: frequency available on EXTINT0 1: frequency available on EXTINT1 2-15: reserved
`fall`	use falling edge of EXTINT pulse (default rising)

6.12.7.7 UBX-MGA-INI-EOP

Message	**UBX-MGA-INI-EOP**					
Description	**Earth Orientation Parameters Assistance**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Input					
Comment	This message allows the delivery of new Earth Orientation Parameters (EOP) to a receiver to improve AssistNow Autonomous operation.					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x13	0x40	72	*see below*	CK_A CK_B
Payload Contents:						

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`type`	-	Message type (0x30 for this type)
1	U1	-	`version`	-	Message version (0x00 for this version)
2	U1[2]	-	`reserved1`	-	Reserved
4	U2	-	`d2kRef`	d	reference time (days since 1.1.2000 12.00h UTC)
6	U2	-	`d2kMax`	d	expiration time (days since 1.1.2000 12.00h UTC)
8	I4	2^-30	`xpP0`	arcsec	x_p t^0 polynomial term (offset)
12	I4	2^-30	`xpP1`	arcsec/d	x_p t^1 polynomial term (drift)
16	I4	2^-30	`ypP0`	arcsec	y_p t^0 polynomial term (offset)
20	I4	2^-30	`ypP1`	arcsec/d	y_p t^1 polynomial term (drift)
24	I4	2^-25	`dUT1`	s	dUT1 t^0 polynomial term (offset)
28	I4	2^-30	`ddUT1`	s/d	dUT1 t^1 polynomial term (drift)
32	U1[40]	-	`reserved2`	-	Reserved

6.12.8 UBX-MGA-QZSS (0x13 0x05)

6.12.8.1 UBX-MGA-QZSS-EPH

Message	**UBX-MGA-QZSS-EPH**					
Description	**QZSS Ephemeris Assistance**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Input					
Comment	This message allows the delivery of QZSS ephemeris assistance to a receiver. See the description of AssistNow Online for details.					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x13	0x05	68	*see below*	CK_A CK_B
Payload Contents:						

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`type`	-	Message type (0x01 for this type)

MGA-QZSS continued

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
1	U1	-	`version`	-	Message version (0x00 for this version)
2	U1	-	`svId`	-	QZSS Satellite identifier (see Satellite Numbering), Range 1-5
3	U1	-	`reserved1`	-	Reserved
4	U1	-	`fitInterval`	-	Fit interval flag
5	U1	-	`uraIndex`	-	URA index
6	U1	-	`svHealth`	-	SV health
7	I1	2^-31	`tgd`	s	Group delay differential
8	U2	-	`iodc`	-	IODC
10	U2	2^4	`toc`	s	Clock data reference time
12	U1	-	`reserved2`	-	Reserved
13	I1	2^-55	`af2`	s/s squared	Time polynomial coefficient 2
14	I2	2^-43	`af1`	s/s	Time polynomial coefficient 1
16	I4	2^-31	`af0`	s	Time polynomial coefficient 0
20	I2	2^-5	`crs`	m	Crs
22	I2	2^-43	`deltaN`	semi-circles/s	Mean motion difference from computed value
24	I4	2^-31	`m0`	semi-circles	Mean anomaly at reference time
28	I2	2^-29	`cuc`	radians	Amp of cosine harmonic corr term to arg of lat
30	I2	2^-29	`cus`	radians	Amp of sine harmonic corr term to arg of lat
32	U4	2^-33	`e`	-	eccentricity
36	U4	2^-19	`sqrtA`	m^0.5	Square root of the semi-major axis A
40	U2	2^4	`toe`	s	Reference time of ephemeris
42	I2	2^-29	`cic`	radians	Amp of cos harmonic corr term to angle of inclination
44	I4	2^-31	`omega0`	semi-circles	Long of asc node of orbit plane at weekly epoch
48	I2	2^-29	`cis`	radians	Amp of sine harmonic corr term to angle of inclination
50	I2	2^-5	`crc`	m	Amp of cosine harmonic corr term to orbit radius
52	I4	2^-31	`i0`	semi-circles	Inclination angle at reference time
56	I4	2^-31	`omega`	semi-circles	Argument of perigee
60	I4	2^-43	`omegaDot`	semi-circles/s	Rate of right ascension
64	I2	2^-43	`idot`	semi-circles/s	Rate of inclination angle
66	U1[2]	-	`reserved3`	-	Reserved

6.12.8.2 UBX-MGA-QZSS-ALM

Message	**UBX-MGA-QZSS-ALM**				
Description	**QZSS Almanac Assistance**				
Firmware	Supported on: - u-blox 9 with protocol version 27				
Type	Input				
Comment	This message allows the delivery of QZSS almanac assistance to a receiver. See the description of AssistNow Online for details.				
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x13	0x05	36	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`type`	-	Message type (0x02 for this type)
1	U1	-	`version`	-	Message version (0x00 for this version)
2	U1	-	`svId`	-	QZSS Satellite identifier (see Satellite Numbering), Range 1-5
3	U1	-	`svHealth`	-	Almanac SV health information
4	U2	2^-21	`e`	-	Almanac eccentricity
6	U1	-	`almWNa`	week	Reference week number of almanac (the 8 bit WNa field)
7	U1	2^12	`toa`	s	Reference time of almanac
8	I2	2^-19	`deltaI`	semi-circles	Delta inclination angle at reference time
10	I2	2^-38	`omegaDot`	semi-circles/s	Almanac rate of right ascension
12	U4	2^-11	`sqrtA`	m^0.5	Almanac square root of the semi-major axis A
16	I4	2^-23	`omega0`	semi-circles	Almanac long of asc node of orbit plane at weekly
20	I4	2^-23	`omega`	semi-circles	Almanac argument of perigee
24	I4	2^-23	`m0`	semi-circles	Almanac mean anomaly at reference time
28	I2	2^-20	`af0`	s	Almanac time polynomial coefficient 0 (8 MSBs)
30	I2	2^-38	`af1`	s/s	Almanac time polynomial coefficient 1
32	U1[4]	-	`reserved1`	-	Reserved

6.12.8.3 UBX-MGA-QZSS-HEALTH

Message	**UBX-MGA-QZSS-HEALTH**					
Description	**QZSS Health Assistance**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Input					
Comment	This message allows the delivery of QZSS health assistance to a receiver. See the description of AssistNow Online for details.					
Message Structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xB5 0x62	0x13	0x05	12	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`type`	-	Message type (0x04 for this type)
1	U1	-	`version`	-	Message version (0x00 for this version)
2	U1[2]	-	`reserved1`	-	Reserved
4	U1[5]	-	`healthCode`	-	Each byte represents a QZSS SV (1-5). The 6 LSBs of each byte contains the 6 bit health code from subframes 4/5, data ID = 3, SV ID = 51
9	U1[3]	-	`reserved2`	-	Reserved

6.13 UBX-MON (0x0A)

Monitoring Messages: i.e. Communication Status, CPU Load, Stack Usage, Task Status.
Messages in the MON class are used to report the receiver status, such as CPU load, stack usage, I/O subsystem statistics etc.

6.13.1 UBX-MON-COMMS (0x0A 0x36)

6.13.1.1 Comm port information

Message	**UBX-MON-COMMS**					
Description	**Comm port information**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Periodic/Polled					
Comment	Consolidated communications information for all ports. The size of the message is determined by the number of ports that are in use on the receiver. A port is only included if communication, either send or receive, has been initiated on that port.					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x0A	0x36	8 + 40*nPorts	*see below*	CK_A CK_B

Payload Contents:

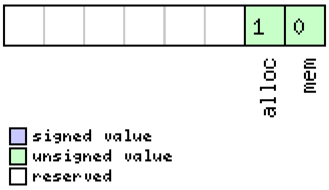
Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`version`	-	Message version (0x00 for this version)
1	U1	-	`nPorts`	-	Number of ports included
2	X1	-	`txErrors`	-	tx error bitmask (see graphic below)
3	U1[1]	-	`reserved1`	-	Reserved
4	U1[4]	-	`protIds`		The identifiers of the protocols reported in the msgs array. 0: UBX, 1: NMEA, 2: RTCM2, 5: RTCM3, 256: No protocol reported.
Start of repeated block (nPorts times)					
8 + 40*N	U2	-	`portId`	-	Unique identifier for the port. See Serial Communications Port Description for details.
10 + 40*N	U2	-	`txPending`	bytes	Number of bytes pending in transmitter buffer
12 + 40*N	U4	-	`txBytes`	bytes	Number of bytes ever sent
16 + 40*N	U1	-	`txUsage`	%	Maximum usage transmitter buffer during the last sysmon period
17 + 40*N	U1	-	`txPeakUsage`	%	Maximum usage transmitter buffer
18 + 40*N	U2	-	`rxPending`	bytes	Number of bytes in receiver buffer
20 + 40*N	U4	-	`rxBytes`	bytes	Number of bytes ever received
24 + 40*N	U1	-	`rxUsage`	%	Maximum usage receiver buffer during the last sysmon period
25 + 40*N	U1	-	`rxPeakUsage`	%	Maximum usage receiver buffer
26 + 40*N	U2	-	`overrunErrs`	-	Number of 100ms timeslots with overrun errors
28 + 40*N	U2[4]	-	`msgs`	msg	Number of successfully parsed messages for each protocol. The reported protocols are identified through the protIds field.
36 + 40*N	U1[8]	-	`reserved2`	-	Reserved
44 + 40*N	U4	-	`skipped`	bytes	Number of skipped bytes

MON-COMMS continued

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
End of repeated block					

Bitfield txErrors

This graphic explains the bits of `txErrors`



Name	*Description*
`mem`	Memory Allocation error
`alloc`	Allocation error (TX buffer full)

6.13.2 UBX-MON-GNSS (0x0A 0x28)

6.13.2.1 Information message major GNSS selection

Message	**UBX-MON-GNSS**					
Description	**Information message major GNSS selection**					
Firmware	Supported on: - u-blox 9 with protocol version 27					
Type	Polled					
Comment	This message reports major GNSS selection. It does this by means of bit masks in U1 fields. Each bit in a bit mask corresponds to one major GNSS. Augmentation systems are not reported.					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x0A	0x28	8	*see below*	CK_A CK_B

Payload Contents:

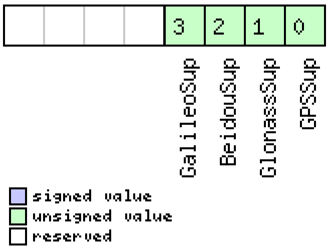
Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`version`	-	Message version (0x01for this version)
1	X1	-	`supported`	-	A bit mask showing the major GNSS that can be supported by this receiver (see graphic below)
2	X1	-	`defaultGnss`	-	A bit mask showing the default major GNSS selection. If the default major GNSS selection is currently configured in the efuse for this receiver, it takes precedence over the default major GNSS selection configured in the executing firmware of this receiver. (see graphic below)
3	X1	-	`enabled`	-	A bit mask showing the current major GNSS selection enabled for this receiver (see graphic below)

MON-GNSS continued

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
4	U1	-	`simultaneous`	-	Maximum number of concurrent major GNSS that can be supported by this receiver
5	U1[3]	-	`reserved1`	-	Reserved

Bitfield supported

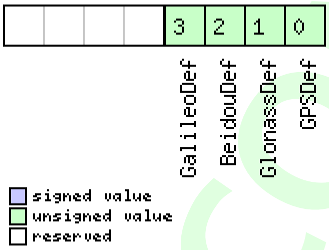
This graphic explains the bits of `supported`



Name	*Description*
`GPSSup`	GPS is supported
`GlonassSup`	GLONASS is supported
`BeidouSup`	BeiDou is supported
`GalileoSup`	Galileo is supported

Bitfield defaultGnss

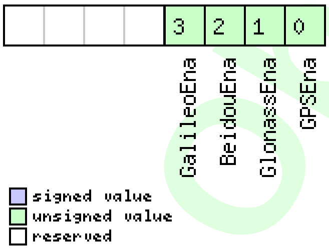
This graphic explains the bits of `defaultGnss`



Name	*Description*
`GPSDef`	GPS is default-enabled
`GlonassDef`	GLONASS is default-enabled
`BeidouDef`	BeiDou is default-enabled
`GalileoDef`	Galileo is default-enabled

Bitfield enabled

This graphic explains the bits of `enabled`



Name	*Description*
`GPSEna`	GPS is enabled
`GlonassEna`	GLONASS is enabled
`BeidouEna`	BeiDou is enabled
`GalileoEna`	Galileo is enabled

6.13.3 UBX-MON-HW2 (0x0A 0x0B)

6.13.3.1 Extended Hardware Status

Message	**UBX-MON-HW2**
Description	**Extended Hardware Status**
Firmware	Supported on: • u-blox 9 with protocol version 27
Type	Periodic/Polled
Comment	**This message is deprecated in this protocol version. Use `UBX-MON-HW3` and `UBX-MON-RF` instead.** Status of different aspects of the hardware such as Imbalance, Low-Level Configuration and POST Results. The first four parameters of this message represent the complex signal from the RF front end. The following rules of thumb apply: • The smaller the absolute value of the variable `ofsI` and `ofsQ`, the better. • Ideally, the magnitude of the I-part (`magI`) and the Q-part (`magQ`) of the complex signal should be the same.

	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x0A	0x0B	28	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	I1	-	`ofsI`	-	Imbalance of I-part of complex signal, scaled (-128 = max. negative imbalance, 127 = max. positive imbalance)
1	U1	-	`magI`	-	Magnitude of I-part of complex signal, scaled (0 = no signal, 255 = max. magnitude)
2	I1	-	`ofsQ`	-	Imbalance of Q-part of complex signal, scaled (-128 = max. negative imbalance, 127 = max. positive imbalance)
3	U1	-	`magQ`	-	Magnitude of Q-part of complex signal, scaled (0 = no signal, 255 = max. magnitude)
4	U1	-	`cfgSource`	-	Source of low-level configuration (114 = ROM, 111 = OTP, 112 = config pins, 102 = flash image)
5	U1[3]	-	`reserved1`	-	Reserved
8	U4	-	`lowLevCfg`	-	Low-level configuration (obsolete in protocol versions greater than 15)
12	U1[8]	-	`reserved2`	-	Reserved
20	U4	-	`postStatus`	-	POST status word
24	U1[4]	-	`reserved3`	-	Reserved

6.13.4 UBX-MON-HW3 (0x0A 0x37)

6.13.4.1 HW I/O pin information

Message	**UBX-MON-HW3**
Description	**HW I/O pin information**
Firmware	Supported on: • u-blox 9 with protocol version 27
Type	Periodic/Polled
Comment	This message contains information specific to each HW I/O pin, for example whether the pin is set as Input or Output. For the antenna supervisor status and other RF status information, see the `UBX-MON-RF` message.

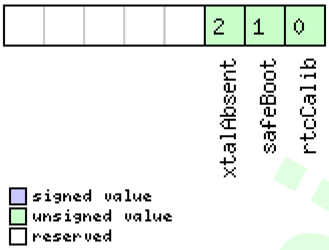
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x0A	0x37	22 + 6*nPins	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`version`	-	Message version (0x00 for this version)
1	U1	-	`nPins`	-	The number of I/O pins included
2	X1	-	`flags`	-	Flags (see graphic below)
3	CH[10]	-	`hwVersion`	-	Zero-terminated Hardware Version String (same as that returned in the `UBX-MON-VER` message)
13	U1[9]	-	`reserved1`	-	Reserved
Start of repeated block (nPins times)					
22 + 6*N	U2	-	`pinId`	-	Identifier for the pin, including both external and internal pins.
24 + 6*N	X2	-	`pinMask`	-	Pin mask (see graphic below)
26 + 6*N	U1	-	`VP`	-	Virtual pin mapping
27 + 6*N	U1	-	`reserved2`	-	Reserved
End of repeated block					

Bitfield flags

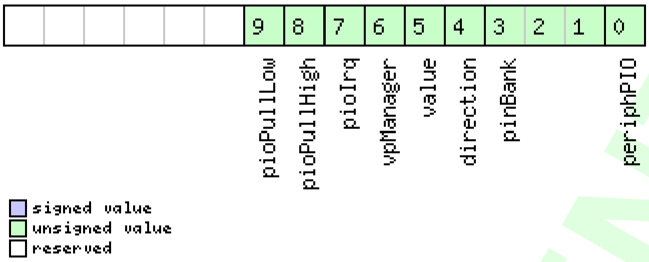
This graphic explains the bits of `flags`



Name	Description
rtcCalib	RTC is calibrated
safeBoot	safeBoot mode (0 = inactive, 1 = active)
xtalAbsent	RTC xtal has been determined to be absent

Bitfield pinMask

This graphic explains the bits of pinMask



Name	Description
periphPIO	Pin is set to peripheral or PIO? 0=Peripheral 1=PIO
pinBank	Bank the pin belongs to, where 0=A 1=B 2=C 3=D 4=E 5=F 6=G 7=H
direction	Pin direction? 0=Input 1=Output
value	Pin value? 0=Low 1=High
vpManager	Used by Virtual Pin Manager? 0=No 1=Yes
pioIrq	Interrupt enabled? 0=No 1=Yes
pioPullHigh	Using Pull High Resistor? 0=No 1=Yes
pioPullLow	Using Pull Low Resistor 0=No 1=Yes

6.13.5 UBX-MON-HW (0x0A 0x09)

6.13.5.1 Hardware Status

Message	**UBX-MON-HW**					
Description	**Hardware Status**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Periodic/Polled					
Comment	**This message is deprecated in this protocol version. Use UBX-MON-HW3 and UBX-MON-RF instead.** Status of different aspect of the hardware, such as Antenna, PIO/Peripheral Pins, Noise Level, Automatic Gain Control (AGC)					
	Header	Class	ID	Length (Bytes)	Payload	Checksum
Message Structure	0xB5 0x62	0x0A	0x09	60	see below	CK_A CK_B

Payload Contents:

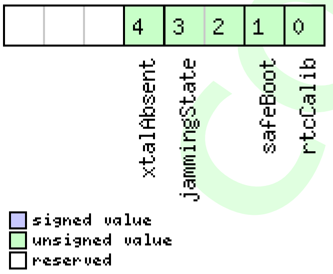
Byte Offset	Number Format	Scaling	Name	Unit	Description
0	X4	-	pinSel	-	Mask of Pins Set as Peripheral/PIO
4	X4	-	pinBank	-	Mask of Pins Set as Bank A/B
8	X4	-	pinDir	-	Mask of Pins Set as Input/Output
12	X4	-	pinVal	-	Mask of Pins Value Low/High
16	U2	-	noisePerMS	-	Noise Level as measured by the GPS Core

MON-HW continued

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
18	U2	-	`agcCnt`	-	AGC Monitor (counts SIGHI xor SIGLO, range 0 to 8191)
20	U1	-	`aStatus`	-	Status of the Antenna Supervisor State Machine (0=INIT, 1=DONTKNOW, 2=OK, 3=SHORT, 4=OPEN)
21	U1	-	`aPower`	-	Current PowerStatus of Antenna (0=OFF, 1=ON, 2=DONTKNOW)
22	X1	-	`flags`	-	Flags (see graphic below)
23	U1	-	`reserved1`	-	Reserved
24	X4	-	`usedMask`	-	Mask of Pins that are used by the Virtual Pin Manager
28	U1[17]	-	`VP`	-	Array of Pin Mappings for each of the 17 Physical Pins
45	U1	-	`jamInd`	-	CW Jamming indicator, scaled (0 = no CW jamming, 255 = strong CW jamming)
46	U1[2]	-	`reserved2`	-	Reserved
48	X4	-	`pinIrq`	-	Mask of Pins Value using the PIO Irq
52	X4	-	`pullH`	-	Mask of Pins Value using the PIO Pull High Resistor
56	X4	-	`pullL`	-	Mask of Pins Value using the PIO Pull Low Resistor

Bitfield flags

This graphic explains the bits of `flags`



Name	*Description*
`rtcCalib`	RTC is calibrated
`safeBoot`	safeBoot mode (0 = inactive, 1 = active)
`jammingState`	output from Jamming/Interference Monitor (0 = unknown or feature disabled, 1 = ok - no significant jamming, 2 = warning - interference visible but fix OK, 3 = critical - interference visible and no fix)
`xtalAbsent`	RTC xtal has been determined to be absent.

6.13.6 UBX-MON-IO (0x0A 0x02)

6.13.6.1 I/O Subsystem Status

Message	**UBX-MON-IO**
Description	**I/O Subsystem Status**
Firmware	Supported on: • u-blox 9 with protocol version 27
Type	Periodic/Polled
Comment	**This message is deprecated in this protocol version. Use UBX-MON-COMMS instead.** The size of the message is determined by the number of ports 'N' the receiver supports, i.e. on u-blox 5 the number of ports is 6.

	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x0A	0x02	0 + 20*N	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
Start of repeated block (N times)					
N*20	U4	-	`rxBytes`	bytes	Number of bytes ever received
4 + 20*N	U4	-	`txBytes`	bytes	Number of bytes ever sent
8 + 20*N	U2	-	`parityErrs`	-	Number of 100ms timeslots with parity errors
10 + 20*N	U2	-	`framingErrs`	-	Number of 100ms timeslots with framing errors
12 + 20*N	U2	-	`overrunErrs`	-	Number of 100ms timeslots with overrun errors
14 + 20*N	U2	-	`breakCond`	-	Number of 100ms timeslots with break conditions
16 + 20*N	U1[4]	-	`reserved1`	-	Reserved
End of repeated block					

6.13.7 UBX-MON-MSGPP (0x0A 0x06)

6.13.7.1 Message Parse and Process Status

Message	**UBX-MON-MSGPP**
Description	**Message Parse and Process Status**
Firmware	Supported on: • u-blox 9 with protocol version 27
Type	Periodic/Polled
Comment	**This message is deprecated in this protocol version. Use UBX-MON-COMMS instead.**

	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x0A	0x06	120	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U2[8]	-	`msg1`	msgs	Number of successfully parsed messages for each protocol on port0
16	U2[8]	-	`msg2`	msgs	Number of successfully parsed messages for each protocol on port1

MON-MSGPP continued

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
32	U2[8]	-	`msg3`	msgs	Number of successfully parsed messages for each protocol on port2
48	U2[8]	-	`msg4`	msgs	Number of successfully parsed messages for each protocol on port3
64	U2[8]	-	`msg5`	msgs	Number of successfully parsed messages for each protocol on port4
80	U2[8]	-	`msg6`	msgs	Number of successfully parsed messages for each protocol on port5
96	U4[6]	-	`skipped`	bytes	Number skipped bytes for each port

6.13.8 UBX-MON-PATCH (0x0A 0x27)

6.13.8.1 Poll Request for installed patches

Message	**UBX-MON-PATCH**					
Description	**Poll Request for installed patches**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Poll Request					
Comment	-					
Message Structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xB5 0x62	0x0A	0x27	0	*see below*	CK_A CK_B
No payload						

6.13.8.2 Output information about installed patches.

Message	**UBX-MON-PATCH**					
Description	**Output information about installed patches.**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Polled					
Comment	-					
Message Structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xB5 0x62	0x0A	0x27	4 + 16*nEntries	*see below*	CK_A CK_B

Payload Contents:

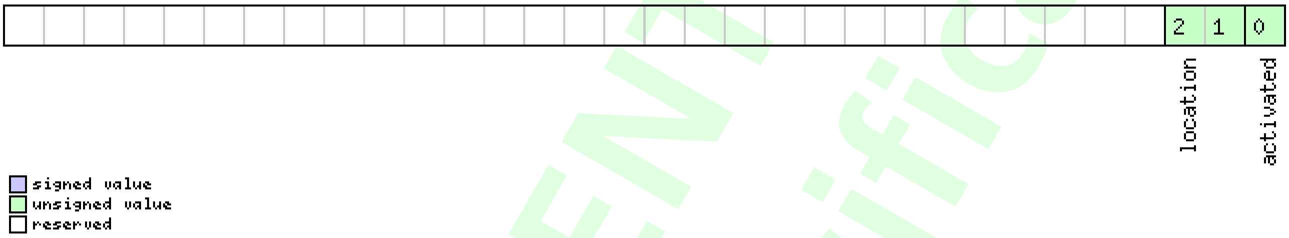
Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U2	-	`version`	-	Type of the message. 0x1 for this one.
2	U2	-	`nEntries`	-	The number of patches that is output.
Start of repeated block (nEntries times)					
4 + 16*N	X4	-	`patchInfo`	-	Additional information about the patch not stated in the patch header. (see graphic below)
8 + 16*N	U4	-	`comparatorNumber`	-	The number of the comparator.
12 + 16*N	U4	-	`patchAddress`	-	The address that the targeted by the patch.

MON-PATCH continued

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
16 + 16*N	U4	-	`patchData`	-	The data that will be inserted at the patchAddress.
End of repeated block					

Bitfield patchInfo

This graphic explains the bits of `patchInfo`



Name	*Description*
`activated`	1: the patch is active. 0: otherwise.
`location`	Indicates where the patch is stored. 0: eFuse, 1: ROM, 2: BBR, 3: file system.

6.13.9 UBX-MON-PIO (0x0A 0x24)

6.13.9.1 Production testing message for PIO pins

Message	**UBX-MON-PIO**
Description	**Production testing message for PIO pins**
Firmware	Supported on: - u-blox 9 with protocol version 27
Type	Polled
Comment	This message is the response to a poll or a result of a `UBX-CFG-PIO` message with an EXTERNAL-NODRIVE or EXTERNAL_DRIVE request. If the message results from a poll, then it will report the PIO pin state. If the receiver has been set in PIO test mode by a `UBX-CFG-PIO` message then this state may have been influenced by SET-PIN forms of the `UBX-CFG-PIO` message. If this message is a response to a EXTERNAL-DRIVE or EXTERNAL-NODRIVE form of `UBX-CFG-PIO` request then the result of an exploration of external pin connection state will be returned instead. The CFG_PIO request will specify which pins are to be examined. Returned state data will be 'unknown' for pins not inspected. If the receiver is running from external SQI memory, tests on PIO0-5, which are the SQI pins, will be blocked. Otherwise the returned state will show states which indicate 'driven low', driven high', 'pulled low', 'pulled high' or 'floating'. For these messages which are a response to an EXTERNAL form of `UBX-CFG-PIO` message, there will be a second set of values following the basic state data. This contains a signal rise or fall time per PIO pin measured in cpu clock cycles. This is only set for pins which have a pull-up or pull-down state and give an extremely approximate idea of rise or fall time. Only a limited range of values can be reported. Each of these is approximately root 2 bigger than the preceding one The final sample is at approximately 128 clocks after the pin ceased to be driven (and starts to move up or down due to the external pull up/down).

	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x0A	0x24	19 + 17	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`version`	-	Message version (0 for this version)
1	U1	-	`responseType`	-	0: Not in test mode, simple read of pin state 1: Test mode, simple read of pin state, which may have been affected by a CFG-PIO SET-PIN request 2: Response to UBX-CFG-PIO EXTERNAL-NODRIVE request 3: Response to UBX-CFG-PIO EXTERNAL-DRIVE request
Start of repeated block (17 times)					
2 + 1*N	U1	-	`pinState`	-	One value per PIO pin: 0: Pin state unknown 1: Pin is floating 2: Pin pulled down 3: Pin pulled up 4: Pin driven high 5: Pin driven low
End of repeated block					
Start of repeated block (17 times)					
19 + 1*N	U1	-	`transitionClocks`	-	One value per PIO pin. The value is a very approximate count of CPU cycles. Zero indicates that there is no transition time measurement for this pin because the transition type is not pull-up or pull-down. The only values that can be reported are 2, 4, 6, 8, 11, 16, 23, 32, 45, 64, 90 and 128. This is only intended to give a general idea of the pull-up or pull-down time constant.
End of repeated block					

6.13.10 UBX-MON-PT2 (0x0A 0x2B)

6.13.10.1 Multi-GNSS Production Test Monitor Message

Message	**UBX-MON-PT2**					
Description	**Multi-GNSS Production Test Monitor Message**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Periodic/Polled					
Comment	-					
Message Structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xB5 0x62	0x0A	0x2B	24 + 28*numRfChn + 36*numSvSigDesc	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`version`	-	Message version (=0 for this version)
1	U1	-	`testMode`	-	currently active test mode
2	U1	-	`numRfChn`	-	number of RF channels reported in this message
3	U1	-	`numSvSigDesc`	-	number of SV signal descriptors reported in this message
4	U4	-	`testRunTime`	ms	test runtime since channel assignment
8	I4	-	`clkDriftAid`	ppb	clock drift of receiver clock relative to extint source (with an offset of 1e9: 1000000000 means 'zero doppler')
12	I4	-	`clkDriftTrk`	ppb	clock drift of receiver clock relative to tracked GNSS signals (without offset: 0 means 'zero doppler')
16	U4	-	`rtcFreq`	Hz	RTC frequency
20	U4	-	`postStatus`	-	Power On Self Test status mask
Start of repeated block (numRfChn times)					
24 + 28*N	U1	-	`rfPga`	-	RF gain amplifier setting
25 + 28*N	U1[27]	-	`reserved1`	-	Reserved
End of repeated block					
Start of repeated block (numSvSigDesc times)					
24 + 28*numRfChn + 36*N	U1	-	`gnssId`	-	GNSS identifier (see Satellite Numbering)
25 + 28*numRfChn + 36*N	U1	-	`svId`	-	GNSS identifier (see Satellite Numbering)
26 + 28*numRfChn + 36*N	U1	-	`sigId`	-	Signal identifier. 0 is the only value currently supported.

MON-PT2 continued

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
27 + 28*numRfChn + 36*N	U1	-	`accsId`	-	Access identifier, used to indicate frequency channel in range (0-13) for GLONASS (0 = -7, 1 = -6, ..., 12 = +5, 13 = +6). The value should be ignored for all other GNSS.
28 + 28*numRfChn + 36*N	U2	2^-8	`cnoMin`	dBHz	minimum CNo across all channels tracking this SV signal
30 + 28*numRfChn + 36*N	U2	2^-8	`cnoMax`	dBHz	maximum CNo across all channels tracking this SV signal
32 + 28*numRfChn + 36*N	U1[14]	-	`reserved2`	-	Reserved
46 + 28*numRfChn + 36*N	U1	2^-8	`carrPhDevMax`	cycles	carrier phase measurement deviation maximum across all associated channels (1 cycle = 360 deg)
47 + 28*numRfChn + 36*N	U1	-	`reserved3`	-	Reserved
48 + 28*numRfChn + 36*N	U1	-	`codeLockSuccess`	%	percentage of channels codelocked
49 + 28*numRfChn + 36*N	U1	-	`phaseLockSuccess`	%	percentage of channels codelocked
50 + 28*numRfChn + 36*N	U2	-	`minCodeLockTime`	ms	minimum codelock time across all associated channels
52 + 28*numRfChn + 36*N	U2	-	`maxCodeLockTime`	ms	maximum codelock time across all associated channels
54 + 28*numRfChn + 36*N	U2	-	`minPhaseLockTime`	ms	minimum phaselock time across all associated channels
56 + 28*numRfChn + 36*N	U2	-	`maxPhaseLockTime`	ms	maximum phaselock time across all associated channels
58 + 28*numRfChn + 36*N	U1[2]	-	`reserved4`	-	Reserved
End of repeated block					

6.13.11 UBX-MON-RF (0x0A 0x38)

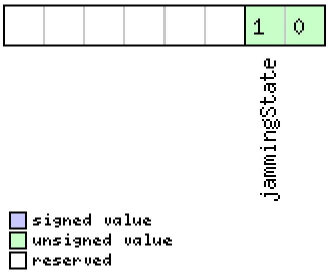
6.13.11.1 RF information

Message	**UBX-MON-RF**					
Description	**RF information**					
Firmware	Supported on: - u-blox 9 with protocol version 27					
Type	Periodic/Polled					
Comment	Information for each RF block.					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x0A	0x38	4 + 24*nBlocks	*see below*	CK_A CK_B

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
Payload Contents:					
0	U1	-	`version`	-	Message version (0x00 for this version)
1	U1	-	`nBlocks`	-	The number of RF blocks included
2	U1[2]	-	`reserved1`	-	Reserved
Start of repeated block (nBlocks times)					
4 + 24*N	U1	-	`blockId`	-	RF block id
5 + 24*N	X1	-	`flags`	-	Flags (see graphic below)
6 + 24*N	U1	-	`antStatus`	-	Status of the antenna supervisor state machine (0x00=INIT,0x01=DONTKNOW,0x02=OK, 0x03=SHORT,0x04=OPEN)
7 + 24*N	U1	-	`antPower`	-	Current power status of antenna (0x00=OFF, 0x01=ON,0x02=DONTKNOW)
8 + 24*N	U4	-	`postStatus`	-	POST status word
12 + 24*N	U1[4]	-	`reserved2`	-	Reserved
16 + 24*N	U2	-	`noisePerMS`	-	Noise level as measured by the GPS core
18 + 24*N	U2	-	`agcCnt`	-	AGC Monitor (counts SIGHI xor SIGLO, range 0 to 8191)
20 + 24*N	U1	-	`jamInd`	-	CW jamming indicator, scaled (0=no CW jamming, 255 = strong CW jamming)
21 + 24*N	I1	-	`ofsI`	-	Imbalance of I-part of complex signal, scaled (-128 = max. negative imbalance, 127 = max. positive imbalance)
22 + 24*N	U1	-	`magI`	-	Magnitude of I-part of complex signal, scaled (0= no signal, 255 = max.magnitude)
23 + 24*N	I1	-	`ofsQ`	-	Imbalance of Q-part of complex signal, scaled (-128 = max. negative imbalance, 127 = max. positive imbalance)
24 + 24*N	U1	-	`magQ`	-	Magnitude of Q-part of complex signal, scaled (0= no signal, 255 = max.magnitude)
25 + 24*N	U1[3]	-	`reserved3`	-	Reserved
End of repeated block					

Bitfield flags

This graphic explains the bits of `flags`



Name	Description
`jammingState`	output from Jamming/Interference Monitor (0 = unknown or feature disabled, 1 = ok - no significant jamming, 2 = warning - interference visible but fix OK, 3 = critical - interference visible and no fix)

6.13.12 UBX-MON-RXBUF (0x0A 0x07)

6.13.12.1 Receiver Buffer Status

Message	**UBX-MON-RXBUF**					
Description	**Receiver Buffer Status**					
Firmware	Supported on: - u-blox 9 with protocol version 27					
Type	Periodic/Polled					
Comment	**This message is deprecated in this protocol version. Use `UBX-MON-COMMS` instead.**					
Message Structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xB5 0x62	0x0A	0x07	24	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U2[6]	-	`pending`	bytes	Number of bytes pending in receiver buffer for each target
12	U1[6]	-	`usage`	%	Maximum usage receiver buffer during the last sysmon period for each target
18	U1[6]	-	`peakUsage`	%	Maximum usage receiver buffer for each target

6.13.13 UBX-MON-RXR (0x0A 0x21)

6.13.13.1 Receiver Status Information

Message	**UBX-MON-RXR**					
Description	**Receiver Status Information**					
Firmware	Supported on: - u-blox 9 with protocol version 27					
Type	Output					
Comment	The receiver ready message is sent when the receiver changes from or to backup mode.					
Message Structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xB5 0x62	0x0A	0x21	1	*see below*	CK_A CK_B

Payload Contents:

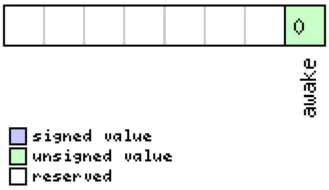
Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*

MON-RXR continued

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	X1	-	`flags`	-	Receiver status flags (see graphic below)

Bitfield flags

This graphic explains the bits of `flags`



Name	*Description*
`awake`	not in Backup mode

6.13.14 UBX-MON-TEMP (0x0A 0x0E)

6.13.14.1 Poll Temperature value [C] and temperature related state

Message	**UBX-MON-TEMP**					
Description	**Poll Temperature value [C] and temperature related state**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Poll Request					
Comment	-					
Message Structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xB5 0x62	0x0A	0x0E	0	*see below*	CK_A CK_B
No payload						

6.13.14.2 Temperature value [C] and temperature related state

Message	**UBX-MON-TEMP**					
Description	**Temperature value [C] and temperature related state**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Periodic/Polled					
Comment	Reports Temperature value [C] and current temperature related state					
Message Structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xB5 0x62	0x0A	0x0E	12	*see below*	CK_A CK_B
Payload Contents:						

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`msgVer`	-	Message version = 0
1	U1[3]	-	`reserved1`	-	Reserved
4	I2	-	`tempValue`	-	Temperature value [C]
6	U1	-	`reserved2`	-	Reserved
7	U1[5]	-	`reserved3`	-	Reserved

6.13.15 UBX-MON-TXBUF (0x0A 0x08)

6.13.15.1 Transmitter Buffer Status

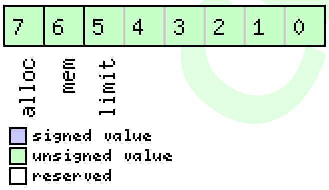
Message	**UBX-MON-TXBUF**					
Description	**Transmitter Buffer Status**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Periodic/Polled					
Comment	**This message is deprecated in this protocol version. Use** UBX-MON-COMMS **instead.**					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x0A	0x08	28	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U2[6]	-	`pending`	bytes	Number of bytes pending in transmitter buffer for each target
12	U1[6]	-	`usage`	%	Maximum usage transmitter buffer during the last sysmon period for each target
18	U1[6]	-	`peakUsage`	%	Maximum usage transmitter buffer for each target
24	U1	-	`tUsage`	%	Maximum usage of transmitter buffer during the last sysmon period for all targets
25	U1	-	`tPeakusage`	%	Maximum usage of transmitter buffer for all targets
26	X1	-	`errors`	-	Error bitmask (see graphic below)
27	U1	-	`reserved1`	-	Reserved

Bitfield errors

This graphic explains the bits of `errors`



Name	*Description*
`limit`	Buffer limit of corresponding target reached
`mem`	Memory Allocation error
`alloc`	Allocation error (TX buffer full)

6.13.16 UBX-MON-VER (0x0A 0x04)

6.13.16.1 Poll Receiver/Software Version

Message	**UBX-MON-VER**					
Description	**Poll Receiver/Software Version**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Poll Request					
Comment	-					
Message Structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xB5 0x62	0x0A	0x04	0	*see below*	CK_A CK_B
No payload						

6.13.16.2 Receiver/Software Version

Message	**UBX-MON-VER**					
Description	**Receiver/Software Version**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Polled					
Comment	-					
Message Structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xB5 0x62	0x0A	0x04	40 + 30*N	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	CH[30]	-	`swVersion`	-	Zero-terminated Software Version String.
30	CH[10]	-	`hwVersion`	-	Zero-terminated Hardware Version String
Start of repeated block (N times)					
40 + 30*N	CH[30]	-	`extension`	-	Extended software information strings. A series of zero-terminated strings. Each extension field is 30 characters long and contains varying software information. Not all extension fields may appear. Example reported information can be: the software version string of the underlying ROM (when the receiver's firmware is running from flash), the firmware version, the supported protocol version, the module identifier, the Flash Information Structure (FIS) file information, the supported major GNSS, the supported augmentation systems.
End of repeated block					

6.14 UBX-NAV (0x01)

Navigation Results Messages: i.e. Position, Speed, Time, Acceleration, Heading, DOP, SVs used.
Messages in the NAV class are used to output navigation data such as position, altitude and velocity in a number of formats. Additionally, status flags and accuracy figures are output. The messages are generated with the configured navigation/measurement rate.

6.14.1 UBX-NAV-CLOCK (0x01 0x22)

6.14.1.1 Clock Solution

Message	**UBX-NAV-CLOCK**					
Description	**Clock Solution**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Periodic/Polled					
Comment	-					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x01	0x22	20	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U4	-	`iTOW`	ms	GPS time of week of the navigation epoch. See the description of iTOW for details.
4	I4	-	`clkB`	ns	Clock bias
8	I4	-	`clkD`	ns/s	Clock drift
12	U4	-	`tAcc`	ns	Time accuracy estimate
16	U4	-	`fAcc`	ps/s	Frequency accuracy estimate

6.14.2 UBX-NAV-COV (0x01 0x36)

6.14.2.1 Covariance matrices

Message	**UBX-NAV-COV**					
Description	**Covariance matrices**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Periodic/Polled					
Comment	This message outputs the covariance matrices for the position and velocity solutions in the topocentric coordinate system defined as the local-level North (N), East (E), Down (D) frame. As the covariance matrices are symmetric, only the upper triangular part is output.					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x01	0x36	64	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U4	-	`iTOW`	ms	GPS time of week of the navigation epoch. See the description of iTOW for details.
4	U1	-	`version`	-	Message version (0 for this version)
5	U1	-	`posCovValid`	-	Position covariance matrix validity flag

NAV-COV continued

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
6	U1	-	`velCovValid`	-	Velocity covariance matrix validity flag
7	U1[9]	-	`reserved1`	-	Reserved
16	R4	-	`posCovNN`	m^2	Position covariance matrix value p_NN
20	R4	-	`posCovNE`	m^2	Position covariance matrix value p_NE
24	R4	-	`posCovND`	m^2	Position covariance matrix value p_ND
28	R4	-	`posCovEE`	m^2	Position covariance matrix value p_EE
32	R4	-	`posCovED`	m^2	Position covariance matrix value p_ED
36	R4	-	`posCovDD`	m^2	Position covariance matrix value p_DD
40	R4	-	`velCovNN`	m^2/s^2	Velocity covariance matrix value v_NN
44	R4	-	`velCovNE`	m^2/s^2	Velocity covariance matrix value v_NE
48	R4	-	`velCovND`	m^2/s^2	Velocity covariance matrix value v_ND
52	R4	-	`velCovEE`	m^2/s^2	Velocity covariance matrix value v_EE
56	R4	-	`velCovED`	m^2/s^2	Velocity covariance matrix value v_ED
60	R4	-	`velCovDD`	m^2/s^2	Velocity covariance matrix value v_DD

6.14.3 UBX-NAV-DOP (0x01 0x04)

6.14.3.1 Dilution of precision

Message	**UBX-NAV-DOP**					
Description	**Dilution of precision**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Periodic/Polled					
Comment	• DOP values are dimensionless. • All DOP values are scaled by a factor of 100. If the unit transmits a value of e.g. 156, the DOP value is 1.56.					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x01	0x04	18	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U4	-	`iTOW`	ms	GPS time of week of the navigation epoch. See the description of iTOW for details.
4	U2	0.01	`gDOP`	-	Geometric DOP
6	U2	0.01	`pDOP`	-	Position DOP
8	U2	0.01	`tDOP`	-	Time DOP
10	U2	0.01	`vDOP`	-	Vertical DOP
12	U2	0.01	`hDOP`	-	Horizontal DOP

NAV-DOP continued

Byte Offset	Number Format	Scaling	Name	Unit	Description
14	U2	0.01	nDOP	-	Northing DOP
16	U2	0.01	eDOP	-	Easting DOP

6.14.4 UBX-NAV-EOE (0x01 0x61)

6.14.4.1 End Of Epoch

Message	**UBX-NAV-EOE**					
Description	**End Of Epoch**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Periodic					
Comment	This message is intended to be used as a marker to collect all navigation messages of an epoch. It is output after all enabled NAV class messages (except NAV-HNR) and after all enabled NMEA messages.					
	Header	Class	ID	Length (Bytes)	Payload	Checksum
Message Structure	0xB5 0x62	0x01	0x61	4	see below	CK_A CK_B

Payload Contents:

Byte Offset	Number Format	Scaling	Name	Unit	Description
0	U4	-	iTOW	ms	GPS time of week of the navigation epoch. See the description of iTOW for details.

6.14.5 UBX-NAV-GEOFENCE (0x01 0x39)

6.14.5.1 Geofencing status

Message	**UBX-NAV-GEOFENCE**					
Description	**Geofencing status**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Periodic/Polled					
Comment	This message outputs the evaluated states of all configured geofences for the current epoch's position. See the Geofencing description for feature details.					
	Header	Class	ID	Length (Bytes)	Payload	Checksum
Message Structure	0xB5 0x62	0x01	0x39	8 + 2*numFences	see below	CK_A CK_B

Payload Contents:

Byte Offset	Number Format	Scaling	Name	Unit	Description
0	U4	-	iTOW	ms	GPS time of week of the navigation epoch. See the description of iTOW for details.
4	U1	-	version	-	Message version (0x00 for this version)
5	U1	-	status	-	Geofencing status 0 - Geofencing not available or not reliable 1 - Geofencing active

NAV-GEOFENCE continued

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
6	U1	-	numFences	-	Number of geofences
7	U1	-	combState	-	Combined (logical OR) state of all geofences 0 - Unknown 1 - Inside 2 - Outside
Start of repeated block (numFences times)					
8 + 2*N	U1	-	state	-	Geofence state 0 - Unknown 1 - Inside 2 - Outside
9 + 2*N	U1[1]	-	reserved1	-	Reserved
End of repeated block					

6.14.6 UBX-NAV-HPPOSECEF (0x01 0x13)

6.14.6.1 High Precision Position Solution in ECEF

Message	**UBX-NAV-HPPOSECEF**					
Description	**High Precision Position Solution in ECEF**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Periodic/Polled					
Comment	**See important comments concerning validity of position given in section Navigation Output Filters.** -					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x01	0x13	28	*see below*	CK_A CK_B
Payload Contents:						

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	version	-	Message version (0 for this version)
1	U1[3]	-	reserved1	-	Reserved
4	U4	-	iTOW	ms	GPS time of week of the navigation epoch. See the description of iTOW for details.
8	I4	-	ecefX	cm	ECEF X coordinate
12	I4	-	ecefY	cm	ECEF Y coordinate
16	I4	-	ecefZ	cm	ECEF Z coordinate
20	I1	0.1	ecefXHp	mm	High precision component of ECEF X coordinate. Must be in the range of -99..+99. Precise coordinate in cm = ecefX + (ecefXHp * 1e-2).
21	I1	0.1	ecefYHp	mm	High precision component of ECEF Y coordinate. Must be in the range of -99..+99. Precise coordinate in cm = ecefY + (ecefYHp * 1e-2).

NAV-HPPOSECEF continued

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
22	I1	0.1	`ecefZHp`	mm	High precision component of ECEF Z coordinate. Must be in the range of -99..+99. Precise coordinate in cm = ecefZ + (ecefZHp * 1e-2).
23	U1	-	`reserved2`	-	Reserved
24	U4	0.1	`pAcc`	mm	Position Accuracy Estimate

6.14.7 UBX-NAV-HPPOSLLH (0x01 0x14)

6.14.7.1 High Precision Geodetic Position Solution

Message	**UBX-NAV-HPPOSLLH**
Description	**High Precision Geodetic Position Solution**
Firmware	Supported on: • u-blox 9 with protocol version 27
Type	Periodic/Polled
Comment	**See important comments concerning validity of position given in section Navigation Output Filters.** This message outputs the Geodetic position with high precision in the currently selected ellipsoid. The default is the WGS84 Ellipsoid, but can be changed with the message `UBX-CFG-DAT`.

	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x01	0x14	36	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`version`	-	Message version (0 for this version)
1	U1[3]	-	`reserved1`	-	Reserved
4	U4	-	`iTOW`	ms	GPS time of week of the navigation epoch. See the description of iTOW for details.
8	I4	1e-7	`lon`	deg	Longitude
12	I4	1e-7	`lat`	deg	Latitude
16	I4	-	`height`	mm	Height above ellipsoid.
20	I4	-	`hMSL`	mm	Height above mean sea level
24	I1	1e-9	`lonHp`	deg	High precision component of longitude. Must be in the range -99..+99. Precise longitude in deg * 1e-7 = lon + (lonHp * 1e-2).
25	I1	1e-9	`latHp`	deg	High precision component of latitude. Must be in the range -99..+99. Precise latitude in deg * 1e-7 = lat + (latHp * 1e-2).
26	I1	0.1	`heightHp`	mm	High precision component of height above ellipsoid. Must be in the range -9..+9. Precise height in mm = height + (heightHp * 0.1).
27	I1	0.1	`hMSLHp`	mm	High precision component of height above mean sea level. Must be in range -9..+9. Precise height in mm = hMSL + (hMSLHp * 0.1)

NAV-HPPOSLLH continued

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
28	U4	0.1	`hAcc`	mm	Horizontal accuracy estimate
32	U4	0.1	`vAcc`	mm	Vertical accuracy estimate

6.14.8 UBX-NAV-ODO (0x01 0x09)

6.14.8.1 Odometer Solution

Message	**UBX-NAV-ODO**				
Description	**Odometer Solution**				
Firmware	Supported on: • u-blox 9 with protocol version 27				
Type	Periodic/Polled				
Comment	This message outputs the traveled distance since last reset (see `UBX-NAV-RESETODO`) together with an associated estimated accuracy and the total cumulated ground distance (can only be reset by a cold start of the receiver).				
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x01	0x09	20	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`version`	-	Message version (0 for this version)
1	U1[3]	-	`reserved1`	-	Reserved
4	U4	-	`iTOW`	ms	GPS time of week of the navigation epoch. See the description of iTOW for details.
8	U4	-	`distance`	m	Ground distance since last reset
12	U4	-	`totalDistance`	m	Total cumulative ground distance
16	U4	-	`distanceStd`	m	Ground distance accuracy (1-sigma)

6.14.9 UBX-NAV-ORB (0x01 0x34)

6.14.9.1 GNSS Orbit Database Info

Message	**UBX-NAV-ORB**					
Description	**GNSS Orbit Database Info**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Periodic/Polled					
Comment	Status of the GNSS orbit database knowledge.					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x01	0x34	8 + 6*numSv	*see below*	CK_A CK_B

Payload Contents:

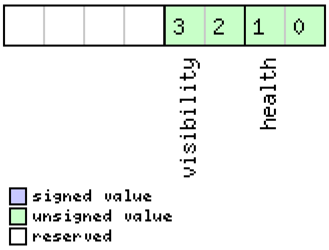
Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U4	-	`iTOW`	ms	GPS time of week of the navigation epoch. See the description of iTOW for details.
4	U1	-	`version`	-	Message version (1, for this version)

NAV-ORB continued

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
5	U1	-	`numSv`	-	Number of SVs in the database
6	U1[2]	-	`reserved1`	-	Reserved
Start of repeated block (numSv times)					
8 + 6*N	U1	-	`gnssId`	-	GNSS ID
9 + 6*N	U1	-	`svId`	-	Satellite ID
10 + 6*N	X1	-	`svFlag`	-	Information Flags (see graphic below)
11 + 6*N	X1	-	`eph`	-	Ephemeris data (see graphic below)
12 + 6*N	X1	-	`alm`	-	Almanac data (see graphic below)
13 + 6*N	X1	-	`otherOrb`	-	Other orbit data available (see graphic below)
End of repeated block					

Bitfield svFlag

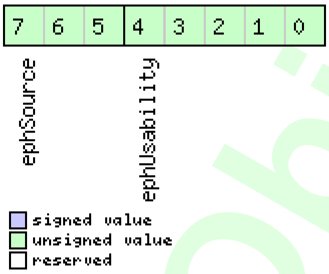
This graphic explains the bits of `svFlag`



Name	*Description*
`health`	SV health: 0: unknown 1: healthy 2: not healty
`visibility`	SV health: 0: unknown 1: below horizon 2: above horizon 3: above elevation mask

Bitfield eph

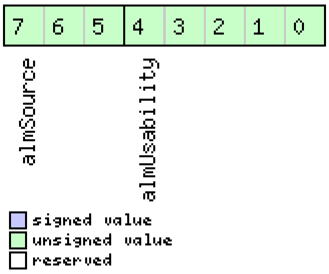
This graphic explains the bits of `eph`



Name	Description
ephUsability	How long the receiver will be able to use the stored ephemeris data from now on: 31: The usability period is unknown 30: The usability period is more than 450 minutes 30 > n > 0: The usability period is between (n-1)*15 and n*15 minutes 0: Ephemeris can no longer be used
ephSource	0: not available 1: GNSS transmission 2: external aiding 3-7: other

Bitfield alm

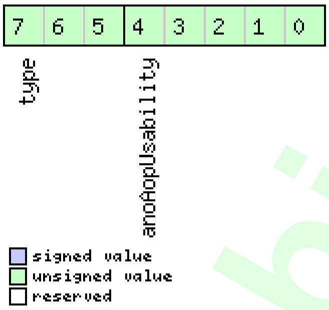
This graphic explains the bits of alm



Name	Description
almUsability	How long the receiver will be able to use the stored almanac data from now on: 31: The usability period is unknown 30: The usability period is more than 30 days 30 > n > 0: The usability period is between n-1 and n days 0: Almanac can no longer be used
almSource	0: not available 1: GNSS transmission 2: external aiding 3-7: other

Bitfield otherOrb

This graphic explains the bits of otherOrb



Name	Description
`anoAopUsability`	How long the receiver will be able to use the orbit data from now on: 31: The usability period is unknown 30: The usability period is more than 30 days 30 > n > 0: The usability period is between n-1 and n days 0: Data can no longer be used
`type`	Type of orbit data: 0: No orbit data available 1: Assist now offline data 2: Assist now autonomous data 3-7: Other orbit data

6.14.10 UBX-NAV-POSECEF (0x01 0x01)

6.14.10.1 Position Solution in ECEF

Message	**UBX-NAV-POSECEF**					
Description	**Position Solution in ECEF**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Periodic/Polled					
Comment	**See important comments concerning validity of position given in section Navigation Output Filters.** -					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x01	0x01	20	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U4	-	`iTOW`	ms	GPS time of week of the navigation epoch. See the description of iTOW for details.
4	I4	-	`ecefX`	cm	ECEF X coordinate
8	I4	-	`ecefY`	cm	ECEF Y coordinate
12	I4	-	`ecefZ`	cm	ECEF Z coordinate
16	U4	-	`pAcc`	cm	Position Accuracy Estimate

6.14.11 UBX-NAV-POSLLH (0x01 0x02)

6.14.11.1 Geodetic Position Solution

Message	**UBX-NAV-POSLLH**
Description	**Geodetic Position Solution**
Firmware	Supported on: - u-blox 9 with protocol version 27
Type	Periodic/Polled
Comment	**See important comments concerning validity of position given in section Navigation Output Filters.** This message outputs the Geodetic position in the currently selected ellipsoid. The default is the WGS84 Ellipsoid, but can be changed with the message `UBX-CFG-DAT`.

	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x01	0x02	28	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U4	-	`iTOW`	ms	GPS time of week of the navigation epoch. See the description of iTOW for details.
4	I4	1e-7	`lon`	deg	Longitude
8	I4	1e-7	`lat`	deg	Latitude
12	I4	-	`height`	mm	Height above ellipsoid
16	I4	-	`hMSL`	mm	Height above mean sea level
20	U4	-	`hAcc`	mm	Horizontal accuracy estimate
24	U4	-	`vAcc`	mm	Vertical accuracy estimate

6.14.12 UBX-NAV-PVT (0x01 0x07)

6.14.12.1 Navigation Position Velocity Time Solution

Message	**UBX-NAV-PVT**
Description	**Navigation Position Velocity Time Solution**
Firmware	Supported on: - u-blox 9 with protocol version 27
Type	Periodic/Polled
Comment	**Note that during a leap second there may be more (or less) than 60 seconds in a minute; see the description of leap seconds for details.** This message combines position, velocity and time solution, including accuracy figures

	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x01	0x07	92	*see below*	CK_A CK_B

Payload Contents:

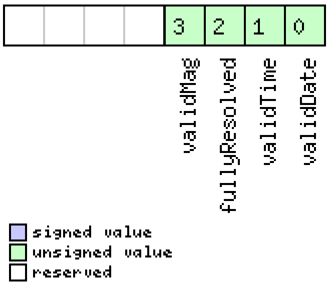
Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U4	-	`iTOW`	ms	GPS time of week of the navigation epoch. See the description of iTOW for details.
4	U2	-	`year`	y	Year (UTC)
6	U1	-	`month`	month	Month, range 1..12 (UTC)
7	U1	-	`day`	d	Day of month, range 1..31 (UTC)

NAV-PVT continued

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
8	U1	-	`hour`	h	Hour of day, range 0..23 (UTC)
9	U1	-	`min`	min	Minute of hour, range 0..59 (UTC)
10	U1	-	`sec`	s	Seconds of minute, range 0..60 (UTC)
11	X1	-	`valid`	-	Validity flags (see graphic below)
12	U4	-	`tAcc`	ns	Time accuracy estimate (UTC)
16	I4	-	`nano`	ns	Fraction of second, range -1e9 .. 1e9 (UTC)
20	U1	-	`fixType`	-	GNSSfix Type: 0: no fix 1: dead reckoning only 2: 2D-fix 3: 3D-fix 4: GNSS + dead reckoning combined 5: time only fix
21	X1	-	`flags`	-	Fix status flags (see graphic below)
22	X1	-	`flags2`	-	Additional flags (see graphic below)
23	U1	-	`numSV`	-	Number of satellites used in Nav Solution
24	I4	1e-7	`lon`	deg	Longitude
28	I4	1e-7	`lat`	deg	Latitude
32	I4	-	`height`	mm	Height above ellipsoid
36	I4	-	`hMSL`	mm	Height above mean sea level
40	U4	-	`hAcc`	mm	Horizontal accuracy estimate
44	U4	-	`vAcc`	mm	Vertical accuracy estimate
48	I4	-	`velN`	mm/s	NED north velocity
52	I4	-	`velE`	mm/s	NED east velocity
56	I4	-	`velD`	mm/s	NED down velocity
60	I4	-	`gSpeed`	mm/s	Ground Speed (2-D)
64	I4	1e-5	`headMot`	deg	Heading of motion (2-D)
68	U4	-	`sAcc`	mm/s	Speed accuracy estimate
72	U4	1e-5	`headAcc`	deg	Heading accuracy estimate (both motion and vehicle)
76	U2	0.01	`pDOP`	-	Position DOP
78	U1[6]	-	`reserved1`	-	Reserved
84	I4	1e-5	`headVeh`	deg	Heading of vehicle (2-D)
88	I2	1e-2	`magDec`	deg	Magnetic declination
90	U2	1e-2	`magAcc`	deg	Magnetic declination accuracy

Bitfield valid

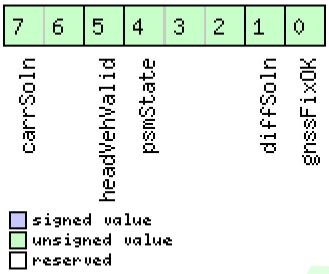
This graphic explains the bits of `valid`



Name	*Description*
`validDate`	1 = valid UTC Date (see Time Validity section for details)
`validTime`	1 = valid UTC Time of Day (see Time Validity section for details)
`fullyResolved`	1 = UTC Time of Day has been fully resolved (no seconds uncertainty)
`validMag`	1 = valid Magnetic declination

Bitfield flags

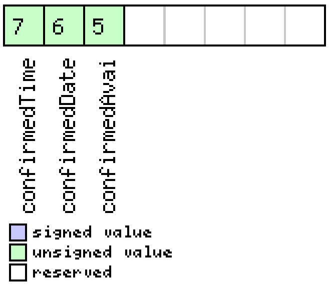
This graphic explains the bits of `flags`



Name	*Description*
`gnssFixOK`	1 = valid fix (i.e within DOP & accuracy masks)
`diffSoln`	1 = differential corrections were applied
`psmState`	Power Save Mode state (see Power Management): 0: PSM is not active 1: Enabled (an intermediate state before Acquisition state 2: Acquisition 3: Tracking 4: Power Optimized Tracking 5: Inactive
`headVehValid`	1 = heading of vehicle is valid
`carrSoln`	Carrier phase range solution status: 0: no carrier phase range solution 1: float solution (no fixed integer carrier phase measurements have been used to calculate the solution) 2: fixed solution (one or more fixed integer carrier phase range measurements have been used to calculate the solution)

Bitfield flags2

This graphic explains the bits of `flags2`



Name	Description
`confirmedAvai`	1 = information about UTC Date and Time of Day validity confirmation is available (see Time Validity section for details). This flag is only supported in Protocol Versions 19.00, 19.10, 20.10, 20.20, 20.30, 22.00, 23.00, 23.01, 27 and 28.
`confirmedDate`	1 = UTC Date validity could be confirmed (see Time Validity section for details)
`confirmedTime`	1 = UTC Time of Day could be confirmed (see Time Validity section for details)

6.14.13 UBX-NAV-RELPOSNED (0x01 0x3C)

6.14.13.1 Relative Positioning Information in NED frame

Message	**UBX-NAV-RELPOSNED**					
Description	**Relative Positioning Information in NED frame**					
Firmware	Supported on: • u-blox 9 with protocol version 27 (**only with High Precision GNSS products**)					
Type	Periodic/Polled					
Comment	**The NED frame is defined as the local topological system at the reference station. The relative position vector components in this message, along with their associated accuracies, are given in that local topological system** This message contains the relative position vector from the Reference Station to the Rover, including accuracy figures, in the local topological system defined at the reference station					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x01	0x3C	40	*see below*	CK_A CK_B
Payload Contents:						

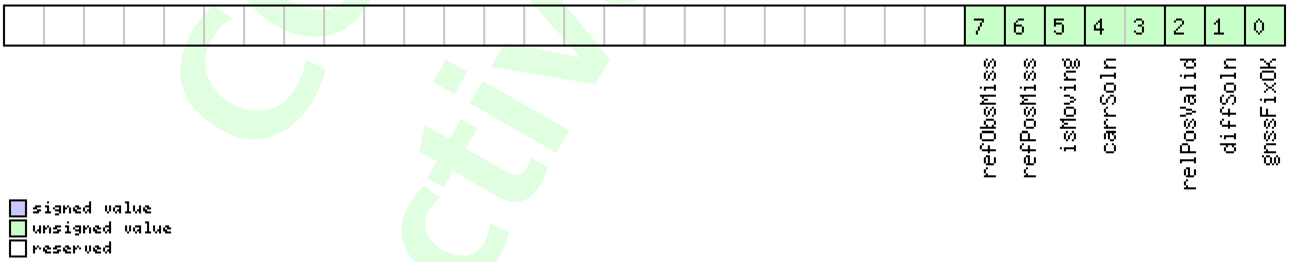
Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`version`	-	Message version (0x00 for this version)
1	U1	-	`reserved1`	-	Reserved
2	U2	-	`refStationId`	-	Reference Station ID. Must be in the range 0..4095
4	U4	-	`iTOW`	ms	GPS time of week of the navigation epoch. See the description of iTOW for details.
8	I4	-	`relPosN`	cm	North component of relative position vector
12	I4	-	`relPosE`	cm	East component of relative position vector
16	I4	-	`relPosD`	cm	Down component of relative position vector

NAV-RELPOSNED continued

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
20	I1	0.1	`relPosHPN`	mm	High-precision North component of relative position vector. Must be in the range -99 to +99. The full North component of the relative position vector, in units of cm, is given by relPosN + (relPosHPN * 1e-2)
21	I1	0.1	`relPosHPE`	mm	High-precision East component of relative position vector. Must be in the range -99 to +99. The full East component of the relative position vector, in units of cm, is given by relPosE + (relPosHPE * 1e-2)
22	I1	0.1	`relPosHPD`	mm	High-precision Down component of relative position vector. Must be in the range -99 to +99. The full Down component of the relative position vector, in units of cm, is given by relPosD + (relPosHPD * 1e-2)
23	U1	-	`reserved2`	-	Reserved
24	U4	0.1	`accN`	mm	Accuracy of relative position North component
28	U4	0.1	`accE`	mm	Accuracy of relative position East component
32	U4	0.1	`accD`	mm	Accuracy of relative position Down component
36	X4	-	`flags`	-	Flags (see graphic below)

Bitfield flags

This graphic explains the bits of `flags`



Name	*Description*
`gnssFixOK`	A valid fix (i.e within DOP & accuracy masks)
`diffSoln`	1 if differential corrections were applied
`relPosValid`	1 if relative position components and accuracies are valid
`carrSoln`	Carrier phase range solution status: 0 = No carrier phase range solution 1 = Float solution. No fixed integer carrier phase measurements have been used to calculate the solution 2 = Fixed solution. One or more fixed integer carrier phase range measurements have been used to calculate the solution
`isMoving`	1 if the receiver is operating in moving baseline mode
`refPosMiss`	1 if extrapolated reference position was used to compute moving baseline solution this epoch

Bitfield flags Description continued

Name	*Description*
`refObsMiss`	1 if extrapolated reference observations were used to compute moving baseline solution this epoch

6.14.14 UBX-NAV-RESETODO (0x01 0x10)

6.14.14.1 Reset odometer

Message	**UBX-NAV-RESETODO**					
Description	**Reset odometer**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Command					
Comment	This message resets the traveled distance computed by the odometer (see `UBX-NAV-ODO`). `UBX-ACK-ACK` or `UBX-ACK-NAK` are returned to indicate success or failure.					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x01	0x10	0	*see below*	CK_A CK_B
No payload						

6.14.15 UBX-NAV-SAT (0x01 0x35)

6.14.15.1 Satellite Information

Message	**UBX-NAV-SAT**					
Description	**Satellite Information**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Periodic/Polled					
Comment	This message displays information about SVs which are either known to be visible or currently tracked by the receiver. All signal related information corresponds to the subset of signals specified in Signal Identifiers.					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x01	0x35	8 + 12*numSvs	*see below*	CK_A CK_B

Payload Contents:

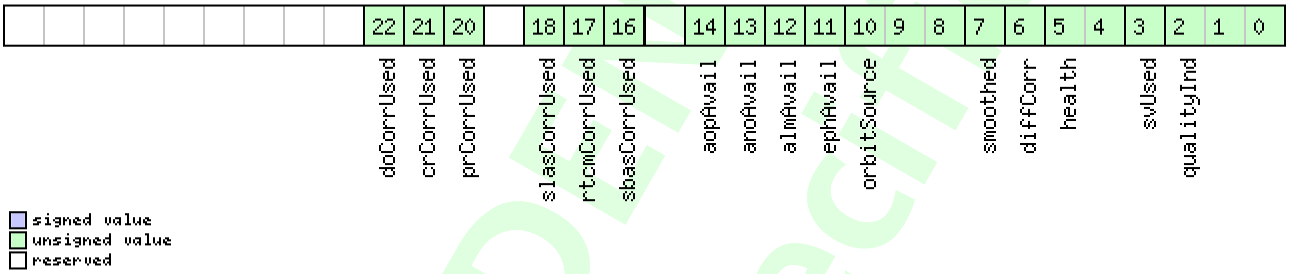
Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U4	-	`iTOW`	ms	GPS time of week of the navigation epoch. See the description of iTOW for details.
4	U1	-	`version`	-	Message version (1 for this version)
5	U1	-	`numSvs`	-	Number of satellites
6	U1[2]	-	`reserved1`	-	Reserved
Start of repeated block (numSvs times)					
8 + 12*N	U1	-	`gnssId`	-	GNSS identifier (see Satellite Numbering) for assignment
9 + 12*N	U1	-	`svId`	-	Satellite identifier (see Satellite Numbering) for assignment
10 + 12*N	U1	-	`cno`	dBHz	Carrier to noise ratio (signal strength)
11 + 12*N	I1	-	`elev`	deg	Elevation (range: +/-90), unknown if out of range

NAV-SAT continued

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
12 + 12*N	I2	-	`azim`	deg	Azimuth (range 0-360), unknown if elevation is out of range
14 + 12*N	I2	0.1	`prRes`	m	Pseudorange residual
16 + 12*N	X4	-	`flags`	-	Bitmask (see graphic below)
End of repeated block					

Bitfield flags

This graphic explains the bits of `flags`



Name	*Description*
`qualityInd`	Signal quality indicator: 0: no signal 1: searching signal 2: signal acquired 3: signal detected but unusable 4: code locked and time synchronized 5, 6, 7: code and carrier locked and time synchronized Note: Since IMES signals are not time synchronized, a channel tracking an IMES signal can never reach a quality indicator value of higher than 3.
`svUsed`	1 = Signal in the subset specified in Signal Identifiers is currently being used for navigation
`health`	Signal health flag: 0: unknown 1: healthy 2: unhealthy
`diffCorr`	1 = differential correction data is available for this SV
`smoothed`	1 = carrier smoothed pseudorange used
`orbitSource`	Orbit source: 0: no orbit information is available for this SV 1: ephemeris is used 2: almanac is used 3: AssistNow Offline orbit is used 4: AssistNow Autonomous orbit is used 5, 6, 7: other orbit information is used
`ephAvail`	1 = ephemeris is available for this SV
`almAvail`	1 = almanac is available for this SV
`anoAvail`	1 = AssistNow Offline data is available for this SV
`aopAvail`	1 = AssistNow Autonomous data is available for this SV

Bitfield flags Description continued

Name	*Description*
`sbasCorrUsed`	1 = SBAS corrections have been used for a signal in the subset specified in Signal Identifiers
`rtcmCorrUsed`	1 = RTCM corrections have been used for a signal in the subset specified in Signal Identifiers
`slasCorrUsed`	1 = QZSS SLAS corrections have been used for a signal in the subset specified in Signal Identifiers
`prCorrUsed`	1 = Pseudorange corrections have been used for a signal in the subset specified in Signal Identifiers
`crCorrUsed`	1 = Carrier range corrections have been used for a signal in the subset specified in Signal Identifiers
`doCorrUsed`	1 = Range rate (Doppler) corrections have been used for a signal in the subset specified in Signal Identifiers

6.14.16 UBX-NAV-SIG (0x01 0x43)

6.14.16.1 Signal Information

Message	**UBX-NAV-SIG**				
Description	**Signal Information**				
Firmware	Supported on: • u-blox 9 with protocol version 27				
Type	Periodic/Polled				
Comment	This message displays information about signals currently tracked by the receiver. On the F9 platform the maximum number of signals is 120.				
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x01	0x43	8 + 16*numSigs	*see below*	CK_A CK_B

Payload Contents:

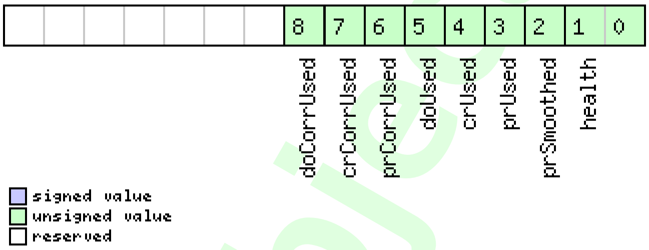
Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U4	-	`iTOW`	ms	GPS time of week of the navigation epoch. See the description of iTOW for details.
4	U1	-	`version`	-	Message version (0x00 for this version)
5	U1	-	`numSigs`	-	Number of signals
6	U1[2]	-	`reserved1`	-	Reserved
Start of repeated block (numSigs times)					
8 + 16*N	U1	-	`gnssId`	-	GNSS identifier (see Satellite Numbering) for assignment
9 + 16*N	U1	-	`svId`	-	Satellite identifier (see Satellite Numbering) for assignment
10 + 16*N	U1	-	`sigId`	-	New style signal identifier (see Signal Identifiers)
11 + 16*N	U1	-	`freqId`	-	Only used for GLONASS: This is the frequency slot + 7 (range from 0 to 13)
12 + 16*N	I2	0.1	`prRes`	m	Pseudorange residual
14 + 16*N	U1	-	`cno`	dBHz	Carrier-to-noise density ratio (signal strength)

NAV-SIG continued

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
15 + 16*N	U1	-	`qualityInd`	-	Signal quality indicator: 0: no signal 1: searching signal 2: signal acquired 3: signal detected but unusable 4: code locked and time synchronized 5, 6, 7: code and carrier locked and time synchronized Note: Since IMES signals are not time synchronized, a channel tracking an IMES signal can never reach a quality indicator value of higher than 3.
16 + 16*N	U1	-	`corrSource`	-	Correction source: 0: no corrections 1: SBAS corrections 2: BeiDou corrections 3: RTCM2 corrections 4: RTCM3 OSR corrections 5: RTCM3 SSR corrections 6: QZSS SLAS corrections
17 + 16*N	U1	-	`ionoModel`	-	Ionospheric model used: 0: no model 1: Klobuchar model transmitted by GPS 2: SBAS model 3: Klobuchar model transmitted by BeiDou
18 + 16*N	X2	-	`sigFlags`	-	Signal related flags (see graphic below)
20 + 16*N	U1[4]	-	`reserved2`	-	Reserved
End of repeated block					

Bitfield sigFlags

This graphic explains the bits of `sigFlags`



Name	Description
health	Signal health flag: 0: unknown 1: healthy 2: unhealthy
prSmoothed	1 = Pseudorange has been smoothed
prUsed	1 = Pseudorange has been used for this signal
crUsed	1 = Carrier range has been used for this signal
doUsed	1 = Range rate (Doppler) has been used for this signal
prCorrUsed	1 = Pseudorange corrections have been used for this signal
crCorrUsed	1 = Carrier range corrections have been used for this signal
doCorrUsed	1 = Range rate (Doppler) corrections have been used for this signal

6.14.17 UBX-NAV-STATUS (0x01 0x03)

6.14.17.1 Receiver Navigation Status

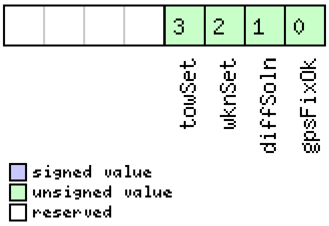
Message	**UBX-NAV-STATUS**					
Description	**Receiver Navigation Status**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Periodic/Polled					
Comment	**See important comments concerning validity of position and velocity given in section Navigation Output Filters.** -					
Message Structure	Header	Class	ID	Length (Bytes)	Payload	Checksum
	0xB5 0x62	0x01	0x03	16	see below	CK_A CK_B

Payload Contents:

Byte Offset	Number Format	Scaling	Name	Unit	Description
0	U4	-	iTOW	ms	GPS time of week of the navigation epoch. See the description of iTOW for details.
4	U1	-	gpsFix	-	GPSfix Type, this value does **not** qualify a fix as valid and within the limits. See note on flag gpsFixOk below. 0x00 = no fix 0x01 = dead reckoning only 0x02 = 2D-fix 0x03 = 3D-fix 0x04 = GPS + dead reckoning combined 0x05 = Time only fix 0x06..0xff = reserved
5	X1	-	flags	-	Navigation Status Flags (see graphic below)
6	X1	-	fixStat	-	Fix Status Information (see graphic below)
7	X1	-	flags2	-	further information about navigation output (see graphic below)
8	U4	-	ttff	ms	Time to first fix (millisecond time tag)
12	U4	-	msss	ms	Milliseconds since Startup / Reset

Bitfield flags

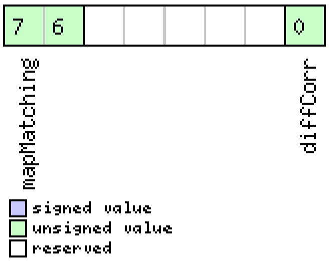
This graphic explains the bits of `flags`



Name	*Description*
`gpsFixOk`	1 = position and velocity valid and within DOP and ACC Masks, see also important comments in section Navigation Output Filters.
`diffSoln`	1 = differential corrections were applied
`wknSet`	1 = Week Number valid (see Time Validity section for details)
`towSet`	1 = Time of Week valid (see Time Validity section for details)

Bitfield fixStat

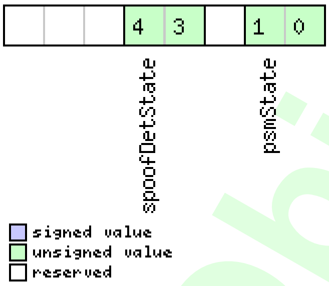
This graphic explains the bits of `fixStat`



Name	*Description*
`diffCorr`	1 = differential corrections available
`mapMatching`	map matching status: 00: none 01: valid but not used, i.e. map matching data was received, but was too old 10: valid and used, map matching data was applied 11: valid and used, map matching data was applied. In case of sensor unavailability map matching data enables dead reckoning. This requires map matched latitude/longitude or heading data.

Bitfield flags2

This graphic explains the bits of `flags2`



Name	*Description*
`psmState`	power save mode state 0: ACQUISITION [or when psm disabled] 1: TRACKING 2: POWER OPTIMIZED TRACKING 3: INACTIVE
`spoofDetState`	Spoofing detection state 0: Unknown or deactivated 1: No spoofing indicated 2: Spoofing indicated 3: Multiple spoofing indications Note that the spoofing state value only reflects the dector state for the current navigation epoch. As spoofing can be detected most easily at the transition from real signal to spoofing signal, this is also where the detector is triggered the most. I.e. a value of *1 - No spoofing indicated* does not mean that the receiver is not spoofed, it simply states that the detector was not triggered in this epoch.

6.14.18 UBX-NAV-SVIN (0x01 0x3B)

6.14.18.1 Survey-in data

Message	**UBX-NAV-SVIN**					
Description	**Survey-in data**					
Firmware	Supported on: - u-blox 9 with protocol version 27 (**only with High Precision GNSS products**)					
Type	Periodic/Polled					
Comment	This message contains information about survey-in parameters.					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x01	0x3B	40	*see below*	CK_A CK_B
Payload Contents:						

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`version`	-	Message version (0x00 for this version)
1	U1[3]	-	`reserved1`	-	Reserved
4	U4	-	`iTOW`	ms	GPS time of week of the navigation epoch. See the description of iTOW for details.
8	U4	-	`dur`	s	Passed survey-in observation time
12	I4	-	`meanX`	cm	Current survey-in mean position ECEF X coordinate
16	I4	-	`meanY`	cm	Current survey-in mean position ECEF Y coordinate
20	I4	-	`meanZ`	cm	Current survey-in mean position ECEF Z coordinate
24	I1	-	`meanXHP`	0.1_ mm	Current high-precision survey-in mean position ECEF X coordinate. Must be in the range -99.. +99. The current survey-in mean position ECEF X coordinate, in units of cm, is given by meanX + (0.01 * meanXHP)

NAV-SVIN continued

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
25	I1	-	`meanYHP`	0.1_ mm	Current high-precision survey-in mean position ECEF Y coordinate. Must be in the range -99..+99. The current survey-in mean position ECEF Y coordinate, in units of cm, is given by meanY + (0.01 * meanYHP)
26	I1	-	`meanZHP`	0.1_ mm	Current high-precision survey-in mean position ECEF Z coordinate. Must be in the range -99..+99. The current survey-in mean position ECEF Z coordinate, in units of cm, is given by meanZ + (0.01 * meanZHP)
27	U1	-	`reserved2`	-	Reserved
28	U4	-	`meanAcc`	0.1_ mm	Current survey-in mean position accuracy
32	U4	-	`obs`	-	Number of position observations used during survey-in
36	U1	-	`valid`	-	Survey-in position validity flag, 1 = valid, otherwise 0
37	U1	-	`active`	-	Survey-in in progress flag, 1 = in-progress, otherwise 0
38	U1[2]	-	`reserved3`	-	Reserved

6.14.19 UBX-NAV-TIMEBDS (0x01 0x24)

6.14.19.1 BDS Time Solution

Message	**UBX-NAV-TIMEBDS**					
Description	**BDS Time Solution**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Periodic/Polled					
Comment	This message reports the precise BDS time of the most recent navigation solution including validity flags and an accuracy estimate.					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x01	0x24	20	*see below*	CK_A CK_B
Payload Contents:						

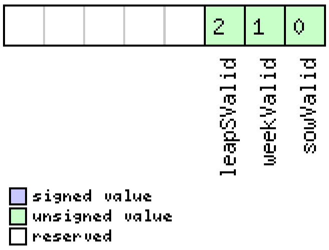
Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U4	-	`iTOW`	ms	GPS time of week of the navigation epoch. See the description of iTOW for details.
4	U4	-	`SOW`	s	BDS time of week (rounded to seconds)
8	I4	-	`fSOW`	ns	Fractional part of SOW (range: +/-500000000). The precise BDS time of week in seconds is: `SOW + fSOW * 1e-9`

NAV-TIMEBDS continued

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
12	I2	-	`week`	-	BDS week number of the navigation epoch
14	I1	-	`leapS`	s	BDS leap seconds (BDS-UTC)
15	X1	-	`valid`	-	Validity Flags (see graphic below)
16	U4	-	`tAcc`	ns	Time Accuracy Estimate

Bitfield valid

This graphic explains the bits of `valid`



Name	*Description*
`sowValid`	1 = Valid SOW and fSOW (see Time Validity section for details)
`weekValid`	1 = Valid week (see Time Validity section for details)
`leapSValid`	1 = Valid leapS

6.14.20 UBX-NAV-TIMEGAL (0x01 0x25)

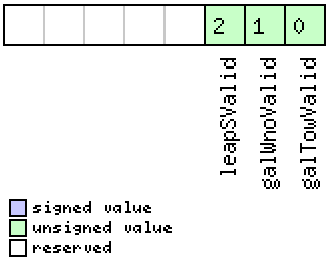
6.14.20.1 Galileo Time Solution

<table>
<tr><td><i>Message</i></td><td colspan="6"><b>UBX-NAV-TIMEGAL</b></td></tr>
<tr><td><i>Description</i></td><td colspan="6"><b>Galileo Time Solution</b></td></tr>
<tr><td><i>Firmware</i></td><td colspan="6">Supported on:
• u-blox 9 with protocol version 27</td></tr>
<tr><td><i>Type</i></td><td colspan="6">Periodic/Polled</td></tr>
<tr><td><i>Comment</i></td><td colspan="6">This message reports the precise Galileo time of the most recent navigation solution including validity flags and an accuracy estimate.</td></tr>
<tr><td rowspan="2"><i>Message Structure</i></td><td><i>Header</i></td><td><i>Class</i></td><td><i>ID</i></td><td><i>Length (Bytes)</i></td><td><i>Payload</i></td><td><i>Checksum</i></td></tr>
<tr><td>0xB5 0x62</td><td>0x01</td><td>0x25</td><td>20</td><td><i>see below</i></td><td>CK_A CK_B</td></tr>
<tr><td colspan="7"><i>Payload Contents:</i></td></tr>
</table>

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U4	-	`iTOW`	ms	GPS time of week of the navigation epoch. See the description of iTOW for details.
4	U4	-	`galTow`	s	Galileo time of week (rounded to seconds)
8	I4	-	`fGalTow`	ns	Fractional part of SOW (range: +/-500000000). The precise Galileo time of week in seconds is: `galTow + fGalTow * 1e-9`
12	I2	-	`galWno`	-	Galileo week number
14	I1	-	`leapS`	s	Galileo leap seconds (Galileo-UTC)
15	X1	-	`valid`	-	Validity Flags (see graphic below)
16	U4	-	`tAcc`	ns	Time Accuracy Estimate

Bitfield valid

This graphic explains the bits of `valid`



Name	Description
`galTowValid`	1 = Valid galTow and fGalTow (see Time Validity section for details)
`galWnoValid`	1 = Valid galWno (see Time Validity section for details)
`leapSValid`	1 = Valid leapS

6.14.21 UBX-NAV-TIMEGLO (0x01 0x23)

6.14.21.1 GLO Time Solution

Message	**UBX-NAV-TIMEGLO**
Description	**GLO Time Solution**
Firmware	Supported on: • u-blox 9 with protocol version 27
Type	Periodic/Polled
Comment	This message reports the precise GLO time of the most recent navigation solution including validity flags and an accuracy estimate.

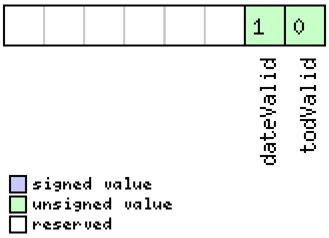
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x01	0x23	20	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U4	-	`iTOW`	ms	GPS time of week of the navigation epoch. See the description of iTOW for details.
4	U4	-	`TOD`	s	GLONASS time of day (rounded to integer seconds)
8	I4	-	`fTOD`	ns	Fractional part of TOD (range: +/-500000000). The precise GLONASS time of day in seconds is: `TOD + fTOD * 1e-9`
12	U2	-	`Nt`	days	Current date (range: 1-1461), starting at 1 from the 1st Jan of the year indicated by N4 and ending at 1461 at the 31st Dec of the third year after that indicated by N4
14	U1	-	`N4`	-	Four-year interval number starting from 1996 (1=1996, 2=2000, 3=2004...)
15	X1	-	`valid`	-	Validity flags (see graphic below)
16	U4	-	`tAcc`	ns	Time Accuracy Estimate

Bitfield valid

This graphic explains the bits of `valid`



Name	Description
`todValid`	1 = Valid TOD and fTOD (see Time Validity section for details)
`dateValid`	1 = Valid N4 and Nt (see Time Validity section for details)

6.14.22 UBX-NAV-TIMEGPS (0x01 0x20)

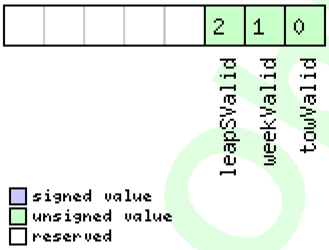
6.14.22.1 GPS Time Solution

Message	**UBX-NAV-TIMEGPS**					
Description	**GPS Time Solution**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Periodic/Polled					
Comment	This message reports the precise GPS time of the most recent navigation solution including validity flags and an accuracy estimate.					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x01	0x20	16	*see below*	CK_A CK_B
Payload Contents:						

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U4	-	`iTOW`	ms	GPS time of week of the navigation epoch. See the description of iTOW for details.
4	I4	-	`fTOW`	ns	Fractional part of iTOW (range: +/-500000). The precise GPS time of week in seconds is: `(iTOW * 1e-3) + (fTOW * 1e-9)`
8	I2	-	`week`	-	GPS week number of the navigation epoch
10	I1	-	`leapS`	s	GPS leap seconds (GPS-UTC)
11	X1	-	`valid`	-	Validity Flags (see graphic below)
12	U4	-	`tAcc`	ns	Time Accuracy Estimate

Bitfield valid

This graphic explains the bits of `valid`



Name	Description
`towValid`	1 = Valid GPS time of week (iTOW & fTOW, see Time Validity section for details)
`weekValid`	1 = Valid GPS week number (see Time Validity section for details)
`leapSValid`	1 = Valid GPS leap seconds

6.14.23 UBX-NAV-TIMELS (0x01 0x26)

6.14.23.1 Leap second event information

Message	**UBX-NAV-TIMELS**					
Description	**Leap second event information**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Periodic/Polled					
Comment	Information about the upcoming leap second event if one is scheduled.					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x01	0x26	24	*see below*	CK_A CK_B

Payload Contents:

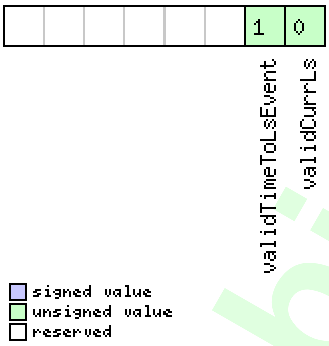
Byte Offset	Number Format	Scaling	Name	Unit	Description
0	U4	-	`iTOW`	ms	GPS time of week of the navigation epoch. See the description of iTOW for details.
4	U1	-	`version`	-	Message version (0x00 for this version).
5	U1[3]	-	`reserved1`	-	Reserved
8	U1	-	`srcOfCurrLs`	-	Information source for the current number of leap seconds. 0: Default (hardcoded in the firmware, can be outdated) 1: Derived from time difference between GPS and GLONASS time 2: GPS 3: SBAS 4: BeiDou 5: Galileo 6: Aided data 7: Configured 255: Unknown
9	I1	-	`currLs`	s	Current number of leap seconds since start of GPS time (Jan 6, 1980). It reflects how much GPS time is ahead of UTC time. Galileo number of leap seconds is the same as GPS. BeiDou number of leap seconds is 14 less than GPS. GLONASS follows UTC time, so no leap seconds.

NAV-TIMELS continued

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
10	U1	-	`srcOfLsChange`	-	Information source for the future leap second event. 0: No source 2: GPS 3: SBAS 4: BeiDou 5: Galileo 6: GLONASS
11	I1	-	`lsChange`	s	Future leap second change if one is scheduled. +1 = positive leap second, -1 = negative leap second, 0 = no future leap second event scheduled or no information available.
12	I4	-	`timeToLsEvent`	s	Number of seconds until the next leap second event, or from the last leap second event if no future event scheduled. If > 0 event is in the future, = 0 event is now, < 0 event is in the past. Valid only if validTimeToLsEvent = 1.
16	U2	-	`dateOfLsGpsWn`	-	GPS week number (WN) of the next leap second event or the last one if no future event scheduled. Valid only if validTimeToLsEvent = 1.
18	U2	-	`dateOfLsGpsDn`	-	GPS day of week number (DN) for the next leap second event or the last one if no future event scheduled. Valid only if validTimeToLsEvent = 1. (GPS and Galileo DN: from 1 = Sun to 7 = Sat. BeiDou DN: from 0 = Sun to 6 = Sat.)
20	U1[3]	-	`reserved2`	-	Reserved
23	X1	-	`valid`	-	Validity flags (see graphic below)

Bitfield valid

This graphic explains the bits of `valid`



Name	*Description*
`validCurrLs`	1 = Valid current number of leap seconds value.
`validTimeToLsEvent`	1 = Valid time to next leap second event or from the last leap second event if no future event scheduled.

6.14.24 UBX-NAV-TIMEUTC (0x01 0x21)

6.14.24.1 UTC Time Solution

Message	**UBX-NAV-TIMEUTC**
Description	**UTC Time Solution**
Firmware	Supported on: • u-blox 9 with protocol version 27
Type	Periodic/Polled
Comment	**Note that during a leap second there may be more or less than 60 seconds in a minute; see the description of leap seconds for details.** -

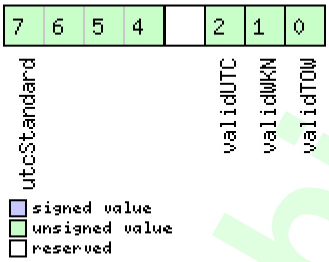
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x01	0x21	20	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U4	-	`iTOW`	ms	GPS time of week of the navigation epoch. See the description of iTOW for details.
4	U4	-	`tAcc`	ns	Time accuracy estimate (UTC)
8	I4	-	`nano`	ns	Fraction of second, range -1e9 .. 1e9 (UTC)
12	U2	-	`year`	y	Year, range 1999..2099 (UTC)
14	U1	-	`month`	month	Month, range 1..12 (UTC)
15	U1	-	`day`	d	Day of month, range 1..31 (UTC)
16	U1	-	`hour`	h	Hour of day, range 0..23 (UTC)
17	U1	-	`min`	min	Minute of hour, range 0..59 (UTC)
18	U1	-	`sec`	s	Seconds of minute, range 0..60 (UTC)
19	X1	-	`valid`	-	Validity Flags (see graphic below)

Bitfield valid

This graphic explains the bits of `valid`



Name	Description
validTOW	1 = Valid Time of Week (see Time Validity section for details)
validWKN	1 = Valid Week Number (see Time Validity section for details)
validUTC	1 = Valid UTC Time
utcStandard	UTC standard identifier. 0: Information not available 1: Communications Research Labratory (CRL) 2: National Institute of Standards and Technology (NIST) 3: U.S. Naval Observatory (USNO) 4: International Bureau of Weights and Measures (BIPM) 5: European Laboratory (tbd) 6: Former Soviet Union (SU) 7: National Time Service Center, China (NTSC) 15: Unknown

6.14.25 UBX-NAV-VELECEF (0x01 0x11)

6.14.25.1 Velocity Solution in ECEF

Message	**UBX-NAV-VELECEF**					
Description	**Velocity Solution in ECEF**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Periodic/Polled					
Comment	**See important comments concerning validity of velocity given in section Navigation Output Filters.** -					
	Header	Class	ID	Length (Bytes)	Payload	Checksum
Message Structure	0xB5 0x62	0x01	0x11	20	see below	CK_A CK_B

Payload Contents:

Byte Offset	Number Format	Scaling	Name	Unit	Description
0	U4	-	iTOW	ms	GPS time of week of the navigation epoch. See the description of iTOW for details.
4	I4	-	ecefVX	cm/s	ECEF X velocity
8	I4	-	ecefVY	cm/s	ECEF Y velocity
12	I4	-	ecefVZ	cm/s	ECEF Z velocity
16	U4	-	sAcc	cm/s	Speed accuracy estimate

6.14.26 UBX-NAV-VELNED (0x01 0x12)

6.14.26.1 Velocity Solution in NED

Message	**UBX-NAV-VELNED**					
Description	**Velocity Solution in NED**					
Firmware	Supported on: - u-blox 9 with protocol version 27					
Type	Periodic/Polled					
Comment	**See important comments concerning validity of velocity given in section Navigation Output Filters.** -					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x01	0x12	36	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U4	-	`iTOW`	ms	GPS time of week of the navigation epoch. See the description of iTOW for details.
4	I4	-	`velN`	cm/s	North velocity component
8	I4	-	`velE`	cm/s	East velocity component
12	I4	-	`velD`	cm/s	Down velocity component
16	U4	-	`speed`	cm/s	Speed (3-D)
20	U4	-	`gSpeed`	cm/s	Ground speed (2-D)
24	I4	1e-5	`heading`	deg	Heading of motion 2-D
28	U4	-	`sAcc`	cm/s	Speed accuracy Estimate
32	U4	1e-5	`cAcc`	deg	Course / Heading accuracy estimate

6.15 UBX-RXM (0x02)

Receiver Manager Messages: i.e. Satellite Status, RTC Status.
Messages in the RXM class are used to output status and result data from the Receiver Manager.

6.15.1 UBX-RXM-MEASX (0x02 0x14)

6.15.1.1 Satellite Measurements for RRLP

Message	**UBX-RXM-MEASX**					
Description	**Satellite Measurements for RRLP**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Periodic					
Comment	The message payload data is, where possible and appropriate, according to the Radio Resource LCS (Location Services) Protocol (RRLP) [1]. One exception is the satellite and GNSS ids, which here are given according to the Satellite Numbering scheme. The correct satellites have to be selected and their satellite ID translated accordingly [1, tab. A.10.14] for use in a RRLP Measure Position Response Component. Similarly, the measurement reference time of week has to be forwarded correctly (modulo 14400000 for the 24 LSB GPS measurements variant, modulo 3600000 for the 22 LSB Galileo and Additional Navigation Satelllite Systems (GANSS) measurements variant) of the RRLP measure position response to the SMLC. Reference: [1] ETSI TS 144 031 V11.0.0 (2012-10), Digital cellular telecommunications system (Phase 2+), Location Services (LCS), Mobile Station (MS) - Serving Mobile Location Centre (SMLC), Radio Resource LCS Protocol (RRLP), (3GPP TS 44.031 version 11.0.0 Release 11).					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x02	0x14	44 + 24*numSV	*see below*	CK_A CK_B

Payload Contents:

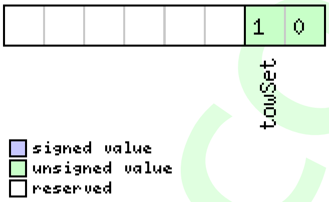
Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`version`	-	Message version, currently 0x00
1	U1[3]	-	`reserved1`	-	Reserved
4	U4	-	`gpsTOW`	ms	GPS measurement reference time
8	U4	-	`gloTOW`	ms	GLONASS measurement reference time
12	U4	-	`bdsTOW`	ms	BeiDou measurement reference time
16	U1[4]	-	`reserved2`	-	Reserved
20	U4	-	`qzssTOW`	ms	QZSS measurement reference time
24	U2	2^-4	`gpsTOWacc`	ms	GPS measurement reference time accuracy (0xffff = > 4s)
26	U2	2^-4	`gloTOWacc`	ms	GLONASS measurement reference time accuracy (0xffff = > 4s)
28	U2	2^-4	`bdsTOWacc`	ms	BeiDou measurement reference time accuracy (0xffff = > 4s)
30	U1[2]	-	`reserved3`	-	Reserved
32	U2	2^-4	`qzssTOWacc`	ms	QZSS measurement reference time accuracy (0xffff = > 4s)
34	U1	-	`numSV`	-	Number of satellites in repeated block

RXM-MEASX continued

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
35	U1	-	`flags`	-	Flags (see graphic below)
36	U1[8]	-	`reserved4`	-	Reserved
Start of repeated block (numSV times)					
44 + 24*N	U1	-	`gnssId`	-	GNSS ID (see Satellite Numbering)
45 + 24*N	U1	-	`svId`	-	Satellite ID (see Satellite Numbering)
46 + 24*N	U1	-	`cNo`	-	carrier noise ratio (0..63)
47 + 24*N	U1	-	`mpathIndic`	-	multipath index (according to [1]) (0 = not measured, 1 = low, 2 = medium, 3 = high)
48 + 24*N	I4	0.04	`dopplerMS`	m/s	Doppler measurement
52 + 24*N	I4	0.2	`dopplerHz`	Hz	Doppler measurement
56 + 24*N	U2	-	`wholeChips`	-	whole value of the code phase measurement (0. .1022 for GPS)
58 + 24*N	U2	-	`fracChips`	-	fractional value of the code phase measurement (0..1023)
60 + 24*N	U4	2^-21	`codePhase`	ms	Code phase
64 + 24*N	U1	-	`intCodePhase`	ms	Integer (part of the) code phase
65 + 24*N	U1	-	`pseuRangeRMSErr`	-	pseudorange RMS error index (according to [1]) (0..63)
66 + 24*N	U1[2]	-	`reserved5`	-	Reserved
End of repeated block					

Bitfield flags

This graphic explains the bits of `flags`



Name	*Description*
`towSet`	TOW set (0 = no, 1 or 2 = yes)

6.15.2 UBX-RXM-PMREQ (0x02 0x41)

6.15.2.1 Requests a Power Management task

Message	**UBX-RXM-PMREQ**
Description	**Requests a Power Management task**
Firmware	Supported on: • u-blox 9 with protocol version 27
Type	Command
Comment	Request of a Power Management related task of the receiver.

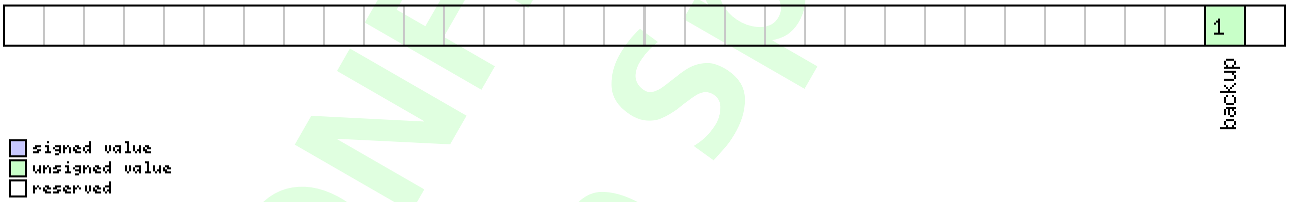
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x02	0x41	8	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U4	-	`duration`	ms	Duration of the requested task, set to zero for infinite duration. The maximum supported time is 12 days.
4	X4	-	`flags`	-	task flags (see graphic below)

Bitfield flags

This graphic explains the bits of `flags`



Name	*Description*
`backup`	The receiver goes into backup mode for a time period defined by duration. Provided that it is not connected to USB

6.15.2.2 Requests a Power Management task

Message	**UBX-RXM-PMREQ**
Description	**Requests a Power Management task**
Firmware	Supported on: • u-blox 9 with protocol version 27
Type	Command
Comment	Request of a Power Management related task of the receiver.

	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x02	0x41	16	*see below*	CK_A CK_B

Payload Contents:

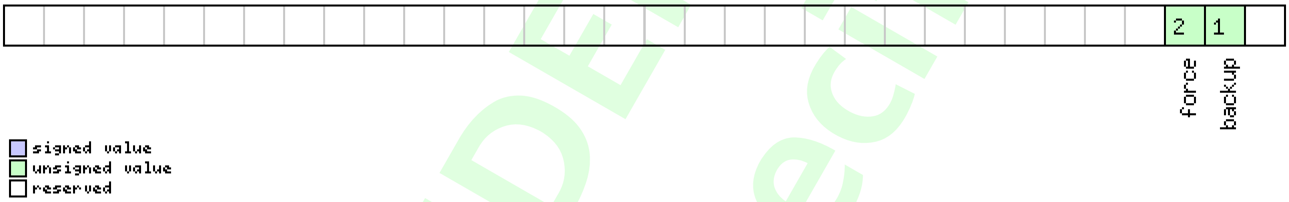
Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`version`	-	Message version (0x00 for this version)
1	U1[3]	-	`reserved1`	-	Reserved

RXM-PMREQ continued

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
4	U4	-	`duration`	ms	Duration of the requested task, set to zero for infinite duration. The maximum supported time is 12 days.
8	X4	-	`flags`	-	task flags (see graphic below)
12	X4	-	`wakeupSources`	-	Configure pins to wakeup the receiver. The receiver wakes up if there is either a falling or a rising edge on one of the configured pins (see graphic below)

Bitfield flags

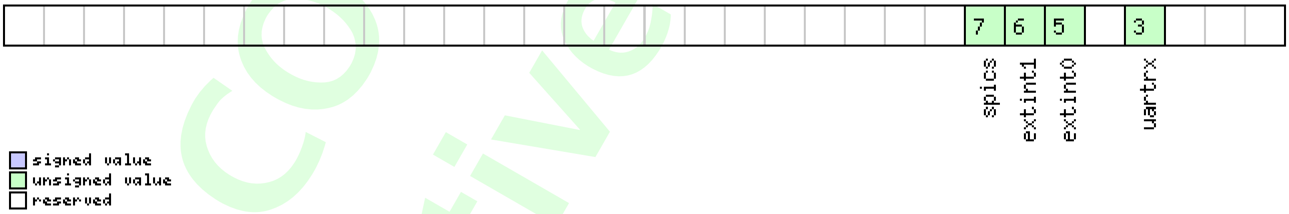
This graphic explains the bits of `flags`



Name	*Description*
`backup`	The receiver goes into backup mode for a time period defined by duration. Provided that it is not connected to USB
`force`	Force receiver backup while USB is connected. USB interface will be disabled.

Bitfield wakeupSources

This graphic explains the bits of `wakeupSources`



Name	*Description*
`uartrx`	Wakeup the receiver if there is an edge on the UART RX pin.
`extint0`	Wakeup the receiver if there is an edge on the EXTINT0 pin.
`extint1`	Wakeup the receiver if there is an edge on the EXTINT1 pin.
`spics`	Wakeup the receiver if there is an edge on the SPI CS pin.

6.15.3 UBX-RXM-RAWX (0x02 0x15)

6.15.3.1 Multi-GNSS Raw Measurement Data

Message	**UBX-RXM-RAWX**					
Description	**Multi-GNSS Raw Measurement Data**					
Firmware	Supported on: • u-blox 9 with protocol version 27 (**only with High Precision GNSS products**)					
Type	Periodic/Polled					
Comment	This message contains the information needed to be able to generate a RINEX 3 multi-GNSS observation file. This message contains pseudorange, Doppler, carrier phase, phase lock and signal quality information for GNSS satellites once signals have been synchronized. This message supports all active GNSS. The only difference between this version of the message and the previous version (**UBX-RXM-RAWX-DATA0**) is the addition of the version field.					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x02	0x15	16 + 32*numMeas	*see below*	CK_A CK_B

Payload Contents:

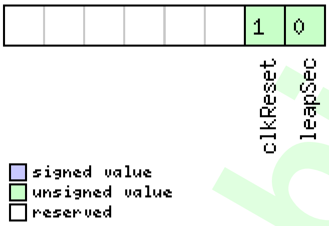
Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	R8	-	`rcvTow`	s	Measurement time of week in receiver local time approximately aligned to the GPS time system. The receiver local time of week, week number and leap second information can be used to translate the time to other time systems. More information about the difference in time systems can be found in RINEX 3 documentation. For a receiver operating in GLONASS only mode, UTC time can be determined by subtracting the leapS field from GPS time regardless of whether the GPS leap seconds are valid.
8	U2	-	`week`	weeks	GPS week number in receiver local time.
10	I1	-	`leapS`	s	GPS leap seconds (GPS-UTC). This field represents the receiver's best knowledge of the leap seconds offset. A flag is given in the recStat bitfield to indicate if the leap seconds are known.
11	U1	-	`numMeas`	-	Number of measurements to follow
12	X1	-	`recStat`	-	Receiver tracking status bitfield (see graphic below)
13	U1	-	`version`	-	Message version (0x01 for this version).
14	U1[2]	-	`reserved1`	-	Reserved
Start of repeated block (numMeas times)					
16 + 32*N	R8	-	`prMes`	m	Pseudorange measurement [m]. GLONASS inter frequency channel delays are compensated with an internal calibration table.

RXM-RAWX continued

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
24 + 32*N	R8	-	`cpMes`	cycles	Carrier phase measurement [cycles]. The carrier phase initial ambiguity is initialized using an approximate value to make the magnitude of the phase close to the pseudorange measurement. Clock resets are applied to both phase and code measurements in accordance with the RINEX specification.
32 + 32*N	R4	-	`doMes`	Hz	Doppler measurement (positive sign for approaching satellites) [Hz]
36 + 32*N	U1	-	`gnssId`	-	GNSS identifier (see Satellite Numbering for a list of identifiers)
37 + 32*N	U1	-	`svId`	-	Satellite identifier (see Satellite Numbering)
38 + 32*N	U1	-	`sigId`	-	New style signal identifier (see Signal Identifiers).
39 + 32*N	U1	-	`freqId`	-	Only used for GLONASS: This is the frequency slot + 7 (range from 0 to 13)
40 + 32*N	U2	-	`locktime`	ms	Carrier phase locktime counter (maximum 64500ms)
42 + 32*N	U1	-	`cno`	dBHz	Carrier-to-noise density ratio (signal strength) [dB-Hz]
43 + 32*N	X1	0.01*2^n	`prStdev`	m	Estimated pseudorange measurement standard deviation (see graphic below)
44 + 32*N	X1	0.004	`cpStdev`	cycles	Estimated carrier phase measurement standard deviation (note a raw value of 0x0F indicates the value is invalid) (see graphic below)
45 + 32*N	X1	0.002*2^n	`doStdev`	Hz	Estimated Doppler measurement standard deviation. (see graphic below)
46 + 32*N	X1	-	`trkStat`	-	Tracking status bitfield (see graphic below)
47 + 32*N	U1	-	`reserved2`	-	Reserved
End of repeated block					

Bitfield recStat

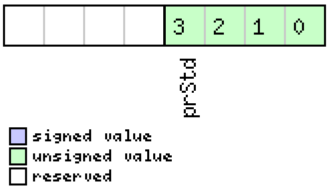
This graphic explains the bits of `recStat`



Name	Description
leapSec	Leap seconds have been determined
clkReset	Clock reset applied. Typically the receiver clock is changed in increments of integer milliseconds.

Bitfield prStdev

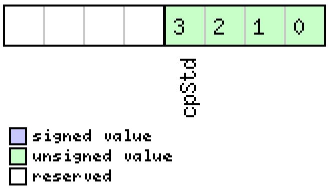
This graphic explains the bits of prStdev



Name	Description
prStd	Estimated pseudorange standard deviation

Bitfield cpStdev

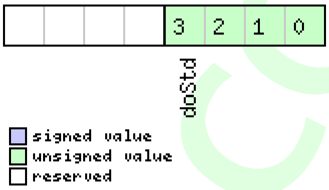
This graphic explains the bits of cpStdev



Name	Description
cpStd	Estimated carrier phase standard deviation

Bitfield doStdev

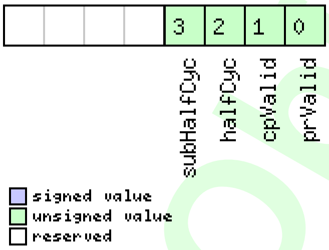
This graphic explains the bits of doStdev



Name	Description
doStd	Estimated Doppler standard deviation

Bitfield trkStat

This graphic explains the bits of trkStat



Name	Description
prValid	Pseudorange valid
cpValid	Carrier phase valid
halfCyc	Half cycle valid
subHalfCyc	Half cycle subtracted from phase

6.15.4 UBX-RXM-RLM (0x02 0x59)

6.15.4.1 Galileo SAR Short-RLM report

Message	**UBX-RXM-RLM**					
Description	**Galileo SAR Short-RLM report**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Output					
Comment	This message contains the contents of any Galileo Search and Rescue (SAR) Short Return Link Message detected by the receiver.					
Message Structure	Header	Class	ID	Length (Bytes)	Payload	Checksum
	0xB5 0x62	0x02	0x59	16	see below	CK_A CK_B

Payload Contents:

Byte Offset	Number Format	Scaling	Name	Unit	Description
0	U1	-	version	-	Message version (0x00 for this version)
1	U1	-	type	-	Message type (0x01 for Short-RLM)
2	U1	-	svId	-	Identifier of transmitting satellite (see Satellite Numbering)
3	U1	-	reserved1	-	Reserved
4	U1[8]	-	beacon	-	Beacon identifier (60 bits), with bytes ordered by earliest transmitted (most significant) first. Top four bits of first byte are zero.
12	U1	-	message	-	Message code (4 bits)
13	U1[2]	-	params	-	Parameters (16 bits), with bytes ordered by earliest transmitted (most significant) first.
15	U1	-	reserved2	-	Reserved

6.15.4.2 Galileo SAR Long-RLM report

Message	**UBX-RXM-RLM**					
Description	**Galileo SAR Long-RLM report**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Output					
Comment	This message contains the contents of any Galileo Search and Rescue (SAR) Long Return Link Message detected by the receiver.					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x02	0x59	28	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`version`	-	Message version (0x00 for this version)
1	U1	-	`type`	-	Message type (0x02 for Long-RLM)
2	U1	-	`svId`	-	Identifier of transmitting satellite (see Satellite Numbering)
3	U1	-	`reserved1`	-	Reserved
4	U1[8]	-	`beacon`	-	Beacon identifier (60 bits), with bytes ordered by earliest transmitted (most significant) first. Top four bits of first byte are zero.
12	U1	-	`message`	-	Message code (4 bits)
13	U1[12]	-	`params`	-	Parameters (96 bits), with bytes ordered by earliest transmitted (most significant) first.
25	U1[3]	-	`reserved2`	-	Reserved

6.15.5 UBX-RXM-RTC5 (0x02 0x23)

6.15.5.1 Real Time Clock Status

Message	**UBX-RXM-RTC5**					
Description	**Real Time Clock Status**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Periodic/Polled					
Comment	Status of the Real time clock					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x02	0x23	28	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U4	-	`rTagHW`	-	RTAG high word
4	U4	-	`rTagLW`	-	RTAG low word
8	U4	-	`freq`	Hz	Clock frequency
12	U4	2^-32	`freqFrac`	Hz	Clock frequency fractional part
16	U4	2^-32	`towFrac`	s	Time of week fractional part
20	U4	-	`tow`	s	Time of week

RXM-RTC5 continued

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
24	U2	-	`wno`	-	GPS week number
26	U1	-	`towValid`	-	TOW is valid flag
27	U1	-	`freqValid`	-	Frequency is valid flag

6.15.6 UBX-RXM-RTCM (0x02 0x32)

6.15.6.1 RTCM input status

Message	**UBX-RXM-RTCM**
Description	**RTCM input status**
Firmware	Supported on: • u-blox 9 with protocol version 27 (**only with High Precision GNSS products**)
Type	Output
Comment	Output upon processing of an RTCM input message

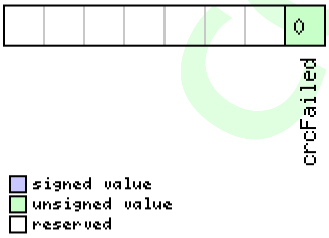
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x02	0x32	8	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`version`	-	Message version (0x02 for this version)
1	X1	-	`flags`	-	RTCM input status flags (see graphic below)
2	U1[2]	-	`reserved1`	-	Reserved
4	U2	-	`refStation`	-	Reference station ID
6	U2	-	`msgType`	-	Message type

Bitfield flags

This graphic explains the bits of `flags`



Name	*Description*
`crcFailed`	0 when RTCM message received and passed CRC check, 1 when failed in which case refStation and msgType might be corrupted and misleading

6.15.7 UBX-RXM-SFRBX (0x02 0x13)

6.15.7.1 Broadcast Navigation Data Subframe

Message	**UBX-RXM-SFRBX**					
Description	**Broadcast Navigation Data Subframe**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Output					
Comment	This message reports a complete subframe of broadcast navigation data decoded from a single signal. The number of data words reported in each message depends on the nature of the signal.					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x02	0x13	8 + 4*numWords	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`gnssId`	-	GNSS identifier (see Satellite Numbering)
1	U1	-	`svId`	-	Satellite identifier (see Satellite Numbering)
2	U1	-	`reserved1`	-	Reserved
3	U1	-	`freqId`	-	Only used for GLONASS: This is the frequency slot + 7 (range from 0 to 13)
4	U1	-	`numWords`	-	The number of data words contained in this message (up to 10, for currently supported signals)
5	U1	-	`chn`	-	The tracking channel number the message was received on
6	U1	-	`version`	-	Message version, (0x02 for this version)
7	U1	-	`reserved2`	-	Reserved
Start of repeated block (numWords times)					
8 + 4*N	U4	-	`dwrd`	-	The data words
End of repeated block					

6.16 UBX-SEC (0x27)

Security Feature Messages

Messages in the SEC class are used for security features of the receiver.

6.16.1 UBX-SEC-SIGN (0x27 0x01)

6.16.1.1 Signature of a previous message

Message	**UBX-SEC-SIGN**					
Description	**Signature of a previous message**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Output					
Comment	The message is the signature of a previously sent message. The signature is generated with a hash using the SHA-256 algorithm with the programmed seeds.					
Message Structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xB5 0x62	0x27	0x01	40	*see below*	CK_A CK_B
Payload Contents:						

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`version`	-	Message version (0x01 for this version)
1	U1[3]	-	`reserved1`	-	Reserved
4	U1	-	`classID`	-	Class ID of the referring message
5	U1	-	`messageID`	-	Message ID of the referring message
6	U2	-	`checksum`	-	UBX Checksum of the referring message
8	U1[32]	-	`hash`	-	SHA-256 hash of the referring message

6.16.2 UBX-SEC-UNIQID (0x27 0x03)

6.16.2.1 Unique Chip ID

Message	**UBX-SEC-UNIQID**					
Description	**Unique Chip ID**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Output					
Comment	This message is used to retrieve a unique chip identifier (40 bits, 5 bytes).					
Message Structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xB5 0x62	0x27	0x03	9	*see below*	CK_A CK_B
Payload Contents:						

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`version`	-	Message version (0x01 for this version)
1	U1[3]	-	`reserved1`	-	Reserved
4	U1[5]	-	`uniqueId`	-	Unique chip ID

6.17 UBX-TIM (0x0D)

Timing Messages: i.e. Time Pulse Output, Time Mark Results.
Messages in the TIM class are used to output timing information from the receiver, like Time Pulse and Time Mark measurements.

6.17.1 UBX-TIM-TM2 (0x0D 0x03)

6.17.1.1 Time mark data

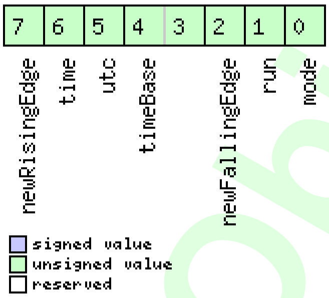
Message	**UBX-TIM-TM2**					
Description	**Time mark data**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Periodic/Polled					
Comment	This message contains information for high precision time stamping / pulse counting. The delay figures and timebase given in `UBX-CFG-TP5` are also applied to the time results output in this message.					
Message Structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xB5 0x62	0x0D	0x03	28	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`ch`	-	Channel (i.e. EXTINT) upon which the pulse was measured
1	X1	-	`flags`	-	Bitmask (see graphic below)
2	U2	-	`count`	-	rising edge counter.
4	U2	-	`wnR`	-	week number of last rising edge
6	U2	-	`wnF`	-	week number of last falling edge
8	U4	-	`towMsR`	ms	tow of rising edge
12	U4	-	`towSubMsR`	ns	millisecond fraction of tow of rising edge in nanoseconds
16	U4	-	`towMsF`	ms	tow of falling edge
20	U4	-	`towSubMsF`	ns	millisecond fraction of tow of falling edge in nanoseconds
24	U4	-	`accEst`	ns	Accuracy estimate

Bitfield flags

This graphic explains the bits of `flags`



Name	*Description*
`mode`	0=single 1=running
`run`	0=armed 1=stopped
`newFallingEdge`	new falling edge detected
`timeBase`	0=Time base is Receiver Time 1=Time base is GNSS Time (the system according to the configuration in `UBX-CFG-TP5` for tpIdx=0) 2=Time base is UTC (the variant according to the configuration in `UBX-CFG-NAV5`)
`utc`	0=UTC not available 1=UTC available
`time`	0=Time is not valid 1=Time is valid (Valid GNSS fix)
`newRisingEdge`	new rising edge detected

6.17.2 UBX-TIM-TP (0x0D 0x01)

6.17.2.1 Time Pulse Timedata

Message	**UBX-TIM-TP**
Description	**Time Pulse Timedata**
Firmware	Supported on: • u-blox 9 with protocol version 27
Type	Periodic/Polled
Comment	This message contains information on the timing of the next pulse at the TIMEPULSE0 output. The recommended configuration when using this message is to set both the measurement rate (`UBX-CFG-RATE`) and the timepulse frequency (`UBX-CFG-TP5`) to 1Hz. TIMEPULSE0 and this message are not available from DR products using the dedicated I2C sensor interface, including NEO-M8L and NEO-M8U modules

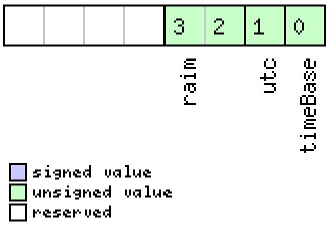
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x0D	0x01	16	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U4	-	`towMS`	ms	Time pulse time of week according to time base
4	U4	2^-32	`towSubMS`	ms	Submillisecond part of TOWMS
8	I4	-	`qErr`	ps	Quantization error of time pulse (not supported for the FTS product variant).
12	U2	-	`week`	weeks	Time pulse week number according to time base
14	X1	-	`flags`	-	bitmask (see graphic below)
15	X1	-	`refInfo`	-	Time reference information (see graphic below)

Bitfield flags

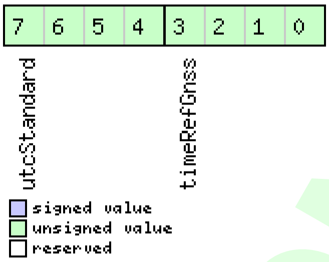
This graphic explains the bits of `flags`



Name	Description
`timeBase`	0=Time base is GNSS 1=Time base is UTC
`utc`	0=UTC not available 1=UTC available
`raim`	(T)RAIM information 0=information not available 1=not active 2=active

Bitfield refInfo

This graphic explains the bits of `refInfo`



Name	Description
`timeRefGnss`	GNSS reference information (only active if time base is GNSS -> timeBase=0) 0: GPS 1: GLONASS 2: BeiDou 15: Unknown
`utcStandard`	UTC standard identifier (only active if time base is UTC -> timeBase=1) 0: Information not available 1: Communications Research Laboratory (CRL) 2: National Institute of Standards and Technology (NIST) 3: U.S. Naval Observatory (USNO) 4: International Bureau of Weights and Measures (BIPM) 5: European Laboratory (tbd) 6: Former Soviet Union (SU) 15: Unknown

6.17.3 UBX-TIM-VRFY (0x0D 0x06)

6.17.3.1 Sourced Time Verification

Message	**UBX-TIM-VRFY**
Description	**Sourced Time Verification**
Firmware	Supported on: • u-blox 9 with protocol version 27
Type	Periodic/Polled
Comment	This message contains verification information about previous time received via AID-INI or from RTC

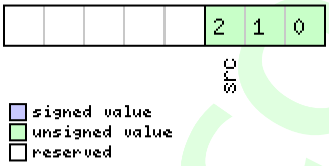
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x0D	0x06	20	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	I4	-	`itow`	ms	integer millisecond tow received by source
4	I4	-	`frac`	ns	sub-millisecond part of tow
8	I4	-	`deltaMs`	ms	integer milliseconds of delta time (current time minus sourced time)
12	I4	-	`deltaNs`	ns	sub-millisecond part of delta time
16	U2	-	`wno`	week	week number
18	X1	-	`flags`	-	information flags (see graphic below)
19	U1	-	`reserved1`	-	Reserved

Bitfield flags

This graphic explains the bits of `flags`



Name	*Description*
`src`	aiding time source 0: no time aiding done 2: source was RTC 3: source was AID-INI

6.18 UBX-UPD (0x09)

Firmware Update Messages: i.e. Memory/Flash erase/write, Reboot, Flash identification, etc..
Messages in the UPD class are used to update the firmware and identify any attached flash device.

6.18.1 UBX-UPD-CERASE (0x09 0x16)

6.18.1.1 Chip erase the connected SQI flash

Message	**UBX-UPD-CERASE**					
Description	**Chip erase the connected SQI flash**					
Firmware	Supported on: - u-blox 9 with protocol version 27					
Type	Command					
Comment	An UBX-ACK-ACK message is sent if the command was received and is valid. A UBX-ACK-NAK is sent if the payload has a wrong size. This response indicates the success state of the command parsing, but does not give an indication whether the command was added to the work queue or had to be discarded because of queue size or memory limitation. If the command could not be added to the queue, the receiver will not provide an additional answer to the UBX-ACK-ACK message. If the command was added to the queue and after flash erase operation was performed, the erase success state is signalized with an UBX-UPD-CERASE output message. Note that depending on the flash it can take up to 5 minutes until the UBX-UPD-CERASE message is output.					
Message Structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xB5 0x62	0x09	0x16	0	*see below*	CK_A CK_B
No payload						

6.18.1.2 Chip erase the connected SQI flash

Message	**UBX-UPD-CERASE**					
Description	**Chip erase the connected SQI flash**					
Firmware	Supported on: - u-blox 9 with protocol version 27					
Type	Output					
Comment	-					
Message Structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xB5 0x62	0x09	0x16	1	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	success	-	1 if success, 0 if chip erase failed

6.18.2 UBX-UPD-CRC (0x09 0x0D)

6.18.2.1 Check CRC over firmware in flash.

Message	**UBX-UPD-CRC**					
Description	**Check CRC over firmware in flash.**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Command					
Comment	**Check of the firmware stored in flash is performed against CRC checksum received in message** An ACK is sent after the answer (UPD-CRC) was sent. A NAK is sent if the payload has a wrong size.					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x09	0x0D	16	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	X4	-	`addr`	-	CRC range begin address
4	U4	-	`size`	-	CRC range size
8	X4	-	`crcA`	-	First word of CRC checksum
12	X4	-	`crcB`	-	Second word of CRC checksum

6.18.2.2 Result of CRC check.

Message	**UBX-UPD-CRC**					
Description	**Result of CRC check.**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Output					
Comment	Contains the result of a CRC check.					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x09	0x0D	5	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	X4	-	`addr`	-	CRC range begin address
4	U1	-	`success`	-	1 if CRC OK, 0 if CRC wrong

6.18.3 UBX-UPD-ERASE (0x09 0x0B)

6.18.3.1 Erase flash sector

Message	**UBX-UPD-ERASE**
Description	**Erase flash sector**
Firmware	Supported on: • u-blox 9 with protocol version 27
Type	Command
Comment	An UBX-ACK-ACK message is sent if the command was received and is valid. A UBX-ACK-NAK is sent if the payload has a wrong size. This response indicates the success state of the command parsing, but does not give an indication whether the command was added to the work queue or had to be discarded because of queue size or memory limitation. If the command could not be added to the queue, the receiver will not provide an additional answer to the UBX-ACK-ACK message. If the command was added to the queue and after flash erase operation was performed, the erase success state is signalized with an UBX-UPD-ERASE output message. Note that depending on the flash it can take up to 10 seconds until the UBX-UPD-ERASE message is output.

	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x09	0x0B	4	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U4	-	`addr`	-	Base address of flash sector

6.18.3.2 Erase flash sector

Message	**UBX-UPD-ERASE**
Description	**Erase flash sector**
Firmware	Supported on: • u-blox 9 with protocol version 27
Type	Output
Comment	-

	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x09	0x0B	5	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	X4	-	`addr`	-	Base address of flash sector
4	U1	-	`success`	-	1 if success, 0 if erase failed

6.18.4 UBX-UPD-FLDET (0x09 0x08)

6.18.4.1 Get the Flash manufacturer and device IDs

Message	**UBX-UPD-FLDET**
Description	**Get the Flash manufacturer and device IDs**
Firmware	Supported on: • u-blox 9 with protocol version 27
Type	Poll Request
Comment	An ACK is sent after the answer (UPD-FLDET with payload) was sent. A NAK is sent if the payload has a wrong size.

	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x09	0x08	4	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U4	-	`address`	-	Base address of Flash

6.18.4.2 Get the Flash manufacturer and device IDs

Message	**UBX-UPD-FLDET**
Description	**Get the Flash manufacturer and device IDs**
Firmware	Supported on: • u-blox 9 with protocol version 27
Type	Get
Comment	This is the response from the receiver

	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x09	0x08	8	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U4	-	`address`	-	Base address of Flash
4	U2	-	`manId`	-	Manufacturer ID
6	U2	-	`devId`	-	Device ID

6.18.5 UBX-UPD-FLWRI (0x09 0x0C)

6.18.5.1 Write flash data (area must be erased before)

Message	**UBX-UPD-FLWRI**				
Description	**Write flash data (area must be erased before)**				
Firmware	Supported on: • u-blox 9 with protocol version 27				
Type	Command				
Comment	An UBX-ACK-ACK message is sent if the command was received and is valid. A UBX-ACK-NAK is sent if the payload has a wrong size or the field 'size' does not match the data payload size. This response indicates the success state of the command parsing, but does not give an indication whether the command was added to the work queue or had to be discarded because of queue size or memory limitation. If the command could not be added to the queue, the receiver will not provide an additional answer to the UBX-ACK-ACK message. If the command was added to the queue and after flash write operation was performed, the write success state is signalized with an UBX-UPD-FLWRI output message.				

Message Structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xB5 0x62	0x09	0x0C	8 + 1*size	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	X4	-	`addr`	-	Base address of write block
4	U4	-	`size`	-	Size of data to write
Start of repeated block (size times)					
8 + 1*N	U1	-	`data`	-	Data to write
End of repeated block					

6.18.5.2 Write flash data success indication

Message	**UBX-UPD-FLWRI**
Description	**Write flash data success indication**
Firmware	Supported on: • u-blox 9 with protocol version 27
Type	Output
Comment	Success report for write command

Message Structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xB5 0x62	0x09	0x0C	5	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	X4	-	`addr`	-	Base address of write block
4	U1	-	`success`	-	1 if success, 0 if write failed

6.18.6 UBX-UPD-IDEN (0x09 0x06)

6.18.6.1 Identify flash loader version

Message	**UBX-UPD-IDEN**					
Description	**Identify flash loader version**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Poll Request					
Comment	An ACK is sent after the answer (UPD-IDEN including payload) was sent. A NAK is sent if the payload has a wrong size.					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x09	0x06	0	*see below*	CK_A CK_B
No payload						

6.18.6.2 Identify flash loader version

Message	**UBX-UPD-IDEN**					
Description	**Identify flash loader version**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Get					
Comment	This is the version response from the receiver.					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x09	0x06	1	*see below*	CK_A CK_B
Payload Contents:						

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	X1	-	`Version`	-	Major.Minor (each 4 Bit)

6.18.7 UBX-UPD-POS (0x09 0x15)

6.18.7.1 Enable PLL during safeboot

Message	**UBX-UPD-POS**					
Description	**Enable PLL during safeboot**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Command					
Comment	The host can send this message in order to enable precise clock. Clock configuration needs to be loaded beforehand using CFG-VALSET message. NAK is sent if the message has wrong size or enabling PLL failed. Host should wait for ACK which is issued after external oscillator and PLL are started It might take a few hundreds of ms before it is completed in case of XTO auto tuning and few dozens of milliseconds in other cases					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x09	0x15	2	*see below*	CK_A CK_B
Payload Contents:						

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*

UPD-POS continued

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`version`	-	Message version (1 for this message)
1	U1	-	`skipOsc`	-	If true OSC initialization is skipped (It has to be already running!)

6.18.8 UBX-UPD-QSIZE (0x09 0x09)

6.18.8.1 Get number of pending commands in queue

Message	**UBX-UPD-QSIZE**					
Description	**Get number of pending commands in queue**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Poll Request					
Comment	Get the number of pending commands in command queue. An ACK is sent after the answer (UPD-QSIZE with payload) was sent. The data will be sent via ANSWER if size is not bigger than 255 and with ANWSER1 otherwise. A NAK is sent if the payload has a wrong size.					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x09	0x09	0	*see below*	CK_A CK_B
No payload						

6.18.8.2 Number of pending commands in queue

Message	**UBX-UPD-QSIZE**					
Description	**Number of pending commands in queue**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Get					
Comment	The number of pending commands in command queue.					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x09	0x09	1	*see below*	CK_A CK_B
Payload Contents:						

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`qSize`	-	Number of commands pending

6.18.8.3 Number of pending commands in queue

Message	**UBX-UPD-QSIZE**					
Description	**Number of pending commands in queue**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Get					
Comment	The number of pending commands in command queue.					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x09	0x09	2	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U2	-	`qSize`	-	Number of commands pending

6.18.9 UBX-UPD-RBOOT (0x09 0x0E)

6.18.9.1 Performs a watchdog reset

Message	**UBX-UPD-RBOOT**					
Description	**Performs a watchdog reset**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Command					
Comment	Performs a watchdog reset after disconnecting USB (if connected). The type of the reset can be compared to a hotstart with an additional operating system reboot. This message is not acknowledged as the system is being reset immediately.					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x09	0x0E	0	*see below*	CK_A CK_B
No payload						

6.18.10 UBX-UPD-ROM (0x09 0x25)

6.18.10.1 Message is holding ROM CRC

Message	**UBX-UPD-ROM**					
Description	**Message is holding ROM CRC**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Polled					
Comment	-					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x09	0x25	12	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`version`	-	Version (Always 0)
1	U1[3]	-	`reserved1`	-	Reserved

UPD-ROM continued

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
4	U4	-	`romCrcLsw`	-	Least significant word of ROM CRC (ROM size - 8 bytes)
8	U4	-	`romCrcMsw`	-	Most significant word of ROM CRC (ROM size - 4 bytes)

6.18.11 UBX-UPD-SAFE (0x09 0x07)

6.18.11.1 Boot in safe environment from ROM or RAM

Message	**UBX-UPD-SAFE**					
Description	**Boot in safe environment from ROM or RAM**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Command					
Comment	Boot receiver in a safe environment from ROM or RAM. An ACK is sent after receiving the command. A NAK is sent if the payload has a wrong size.					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x09	0x07	0	*see below*	CK_A CK_B
No payload						

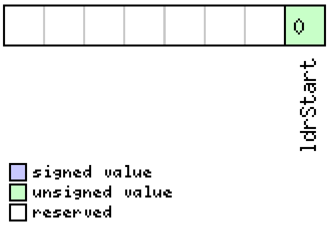
6.18.11.2 Start flash loader task

Message	**UBX-UPD-SAFE**					
Description	**Start flash loader task**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Command					
Comment	If already running the firmware from ROM, the Flash loader task has to be started prior to sending update messages (especially flash-write and erase). The receiver does not need to be started in safe environment. An ACK is sent after receiving the command. A NAK is sent if the payload has a wrong size.					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x09	0x07	1	*see below*	CK_A CK_B
Payload Contents:						

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	X1	-	`flags`	-	flags (see graphic below)

Bitfield flags

This graphic explains the bits of `flags`



Name	Description
`ldrStart`	start flash loader task (flash write and erase routines)

6.18.12 UBX-UPD-SETQ (0x09 0x0F)

6.18.12.1 Set maximum of pending commands in queue

Message	**UBX-UPD-SETQ**					
Description	**Set maximum of pending commands in queue**					
Firmware	Supported on: - u-blox 9 with protocol version 27					
Type	Set					
Comment	Set the maximum number of pending commands in command queue. An ACK is sent after receiving the command. A NACK is sent if the payload has wrong size.					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x09	0x0F	1	*see below*	CK_A CK_B
Payload Contents:						

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`qSize`	-	Number of commands allowed to be pending

6.18.12.2 Set maximum of pending commands in queue

Message	**UBX-UPD-SETQ**					
Description	**Set maximum of pending commands in queue**					
Firmware	Supported on: - u-blox 9 with protocol version 27					
Type	Set					
Comment	Set the maximum number of pending commands in command queue. An ACK is sent after receiving the command. A NACK is sent if the payload has wrong size.					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x09	0x0F	2	*see below*	CK_A CK_B
Payload Contents:						

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U2	-	`qSize`	-	Number of commands allowed to be pending

6.18.13 UBX-UPD-SOS (0x09 0x14)

6.18.13.1 Poll Backup File Restore Status

Message	**UBX-UPD-SOS**					
Description	**Poll Backup File Restore Status**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Poll Request					
Comment	Sending this (empty / no-payload) message to the receiver results in the receiver returning a *System Restored from Backup* message as defined below.					
Message Structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xB5 0x62	0x09	0x14	0	*see below*	CK_A CK_B
No payload						

6.18.13.2 Create Backup File in Flash

Message	**UBX-UPD-SOS**					
Description	**Create Backup File in Flash**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Command					
Comment	The host can send this message in order to save part of the BBR memory in a file in flash file system. The feature is designed in order to emulate the presence of the backup battery even if it is not present; the host can issue the save on shutdown command before switching off the device supply. It is recommended to issue a GNSS stop command before, in order to keep the BBR memory content consistent.					
Message Structure	*Header*	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
	0xB5 0x62	0x09	0x14	4	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`cmd`	-	Command (must be 0)
1	U1[3]	-	`reserved1`	-	Reserved

6.18.13.3 Clear Backup in Flash

Message	**UBX-UPD-SOS**					
Description	**Clear Backup in Flash**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Command					
Comment	The host can send this message in order to erase the backup file present in flash. It is recommended that the clear operation is issued after the host has received the notification that the memory has been restored after a reset. Alternatively the host can parse the startup string 'Restored data saved on shutdown' or poll the UBX-UPD-SOS message for getting the status.					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x09	0x14	4	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`cmd`	-	Command (must be 1)
1	U1[3]	-	`reserved1`	-	Reserved

6.18.13.4 Backup File Creation Acknowledge

Message	**UBX-UPD-SOS**					
Description	**Backup File Creation Acknowledge**					
Firmware	Supported on: • u-blox 9 with protocol version 27					
Type	Output					
Comment	The message is sent from the device as confirmation of creation of a backup file in flash. The host can safely shut down the device after received this message.					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x09	0x14	8	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`cmd`	-	Command (must be 2)
1	U1[3]	-	`reserved1`	-	Reserved
4	U1	-	`response`	-	0: Not acknowledged 1: Acknowledged
5	U1[3]	-	`reserved2`	-	Reserved

6.18.13.5 System Restored from Backup

Message	**UBX-UPD-SOS**					
Description	**System Restored from Backup**					
Firmware	Supported on: - u-blox 9 with protocol version 27					
Type	Output					
Comment	The message is sent from the device to notify the host the BBR has been restored from a backup file in flash. The host should clear the backup file after receiving this message. If the UBX-UPD-SOS message is polled, this message will be resent.					
	Header	*Class*	*ID*	*Length (Bytes)*	*Payload*	*Checksum*
Message Structure	0xB5 0x62	0x09	0x14	8	*see below*	CK_A CK_B

Payload Contents:

Byte Offset	*Number Format*	*Scaling*	*Name*	*Unit*	*Description*
0	U1	-	`cmd`	-	Command (must be 3)
1	U1[3]	-	`reserved1`	-	Reserved
4	U1	-	`response`	-	0: Unknown 1: Failed restoring from backup file 2: Restored from backup file 3: Not restored (no backup)
5	U1[3]	-	`reserved2`	-	Reserved

7 CFG Interface

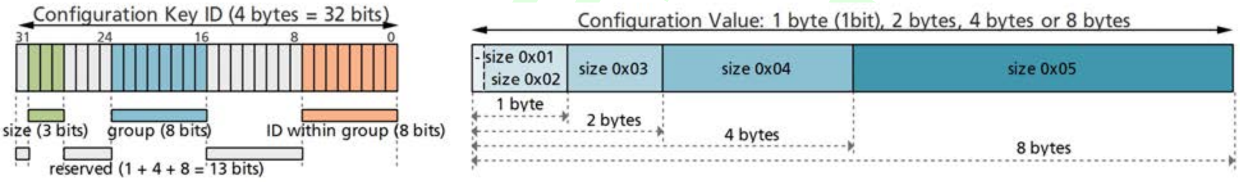
This chapter describes the Receiver Configuration Database accessible thorough the Configuration Interface.

7.1 Configuration Database

The configuration database in the receiver's RAM holds the current configuration, which is used by the receiver at run-time. It is constructed on startup of the receiver from several sources of configuration. These sources are called *Configuration Layers*. The current configuration is called the *RAM Layer*. Any configuration in any layer is organised as *Configuration Items*, which are referenced by a unique *Configuration Key ID* and hold a single *Configuration Value*.

7.2 Configuration Items

The following figure shows the structure of a *Configuration Item*, which consists of a *(Configuration) Key ID* and its *(Configuration) Value*:


Configuration Key ID (4 bytes = 32 bits)
31
24
16
8
0
size (3 bits)
group (8 bits)
ID within group (8 bits)
reserved (1 + 4 + 8 = 13 bits)
Configuration Value: 1 byte (1bit), 2 bytes, 4 bytes or 8 bytes
size 0x01
size 0x02
size 0x03
size 0x04
size 0x05
1 byte
2 bytes
4 bytes
8 bytes

A Configuration Key ID is a 32 bits integer value, which is split into three parts (Note that bits 31, 27..24 and 15..8 are reserved for future use and are currently unused.):

- bits 30..28: 3 bits that indicate the storage size of a Configuration Value (range 0x01-0x05, see below)
- bits 23..16: 8 bits that define a unique group ID (range 0x01-0xfe)
- bits 7..0: 8 bits that define a unique item ID within a group (range 0x01-0xfe)

The entire 32 bits value is the unique Key ID, which uniquely identifies a particular item. The numeric representation of the ID uses the lower-case hexadecimal format, such as `0x20c400a1`. An easier, more readable text representation uses the form CFG-*GROUP-ITEM*. This is also referred to as the *(Configuration) Key Name*.

The storage size identifiers (bits 30..28 of the Key ID) are:

- 0x01: one bit (the actual storage used is one byte, but only the least significant bit is used)
- 0x02: one byte
- 0x03: two bytes
- 0x04: four bytes
- 0x05: eight bytes

Each Configuration Item is of a certain type, which defines the interpretation of the raw binary data (see also number formats):

- U1, U2, U4, U8: unsigned little-endian integers of 8-, 16-, 32- and 64-bit widths
- I1, I2, I4, I8: signed little-endian, two's complement integers of 8-, 16-, 32- and 64-bit widths
- R4, R8: IEEE754 single (32-bit) and double (64-bit) precision floats
- E1, E2, E4: unsigned little-endian enumeration of 8-, 16-, and 32-bit widths (like U1, U2 and U4)
- X1, X2, X4, X8: unsigned little-endian integers of 8-, 16-, 32- and 64-bit widths for bitfields and other binary data, such as strings

- L: single-bit boolean (true = 1, false = 0), stored as U1

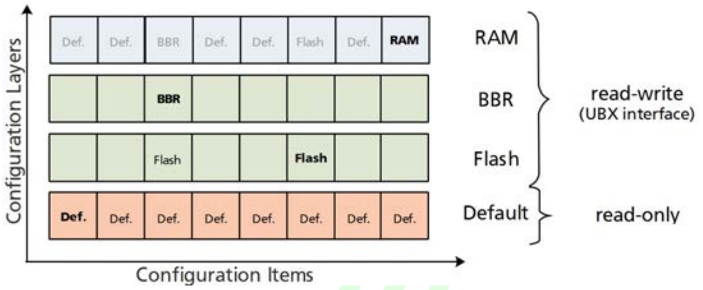
7.3 Configuration Layers

Several *Configuration Layers* exist. They are separate sources of Configuration Items. Some of the layers are read-only and others are modifiable. Layers are organised in terms of priority. Values in a high priority layer will replace values stored in low priority layer. On startup of the receiver all configuration layers are read and the items within each layer are stacked up in order to create the *Current Configuration*, which is used by the receiver at run-time.

The following configuration layers are available (in order of priority, highest priority first):

- **RAM**: This layer contains items stored in volatile RAM. This is the Current Configuration. The value of any item can be set by the user at run-time (see UBX Protocol Interface below) and it will become effective immediately.
- **BBR**: This layer contains items stored in the battery-backed RAM. The contents in this layer are preserved as long as a battery backup supply is provided during off periods. The value of any item can be set by the user at run-time (see UBX Protocol Interface below) and it will become effective upon a restart of the receiver.
- **Flash**: This layer contains items stored permanently in the external flash memory. This layer is only available if there is a usable external flash memory. The value of any item can be set by the user at run-time (see UBX Protocol Interface below) it will become effective upon a restart of the receiver.
- **Default:** This layer contains all items known to the running receiver software and their hard-coded default values. Data in this layer is not writable.

The stacking of the Configuration items from the different layers (sources) in order to construct the Current Configuration in the RAM Layer is depicted in the following figure. For each defined item, i.e. for each item in the Default Layer, the receiver software goes through the layers above and stacks found items on top. Some items may or may not be present in some layers. The result is the RAM Layer filled with all defined items and values coming from the highest priority layer the corresponding item was present. Bold text indicates the source of the value in the Current Configuration (the RAM Layer). Empty boxes mean that the layer can hold the item but that it is not currently stored there. Boxes with text mean that an item is currently stored in the layer.



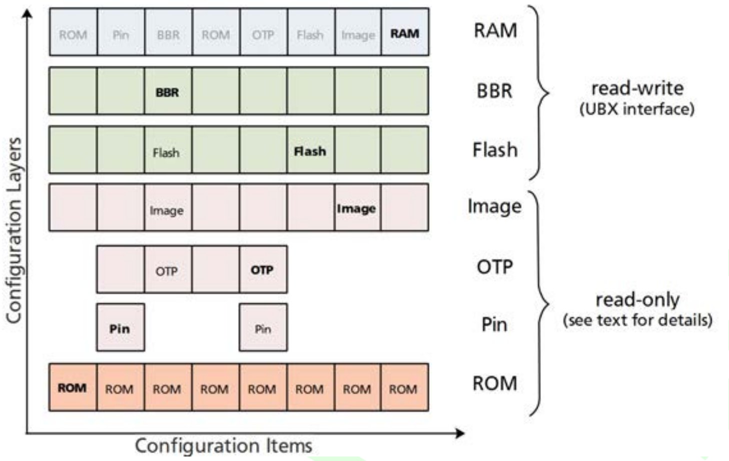
In the example figure above several items (e.g. the first item) are only set in the Default Layer and hence the default value ends up in Current Configuration in the RAM Layer. The third item is present in the Default, Flash and BBR Layers. The value from the BBR Layer has the highest priority and therefore it ends up in the RAM Layer. On the other hand, the default value of the sixth item is changed by the value in the Flash Layer. The value of the last item is changed in the RAM Layer only, i.e. upon startup the value in the RAM Layer was the value from the Default Layer, but the user has changed the value in the RAM Layer at run-time.

7.3.1 Default Layer Composite

That the Default Layer is a composite of the following four layers. Some of these are writable by special means different from the UBX Protocol Interface used for the read-write layers listed above.

- **Image:** This layer contains items appended to an external flash firmware image. It is not modifiable by the user. It is used to provide firmware images that differ in their default (factory) configuration but not in their software or the items in the ROM layer.
- **OTP:** This layer contains items from the contents of the eFuse OTP (one-time programmable [memory]). See OTP Layer Configuration for details.
- **Pin:** This layer contains items derived from configuration pins. See Pin Layer Configuration for details.
- **ROM:** This layer defines all items known to the running receiver software and their hard-coded default value. Data in this layer is not writeable.

The figure below shows all seven layers. An empty space indicates that the item cannot be stored in that layer.



In the example figure above, the first and fourth items are only present in the ROM Layer. Hence the value from the ROM Layer ends up in the RAM Layer. The second item is also present in the Pin Layer and hence that value ends up in the RAM Layer. The third item is present in the ROM, OTP, Image, Flash and BBR Layers. Since the BBR Layer has the highest priority, this value will end up in the RAM Layer. The seventh item is present in the ROM and Image Layers. There is no corresponding item in the Flash or BBR Layers and so the value from the Image Layer ends up in the RAM Layer. The last item is present in the ROM and the RAM Layers. Upon startup the value in the RAM Layer was the value from the ROM Layer. But here the user has changed the value in the RAM Layer at run-time.

7.4 Configuration Interface Access

The following sections describe the existing interfaces to access the Configuration Database.

7.4.1 UBX Protocol Interface

The following UBX protocol messages are available to access the Configuration Database:

- `UBX-CFG-VALGET` to read Configuration Items from the database
- `UBX-CFG-VALSET` to set Configuration Items in the database
- `UBX-CFG-VALDEL` to delete Configuration Items from the database

7.4.2 Pin Layer Configuration

Some Configuration Items are available in the Pin Layer. See the *Hardware Integration Manual*, section *Configuration Pins* for details on how to use configuration pins and how their state affects the values of these items in the Pin Layer.

See also Pin Layer Configuration in the Receiver Description chapter of this document.

7.4.3 OTP Layer Configuration

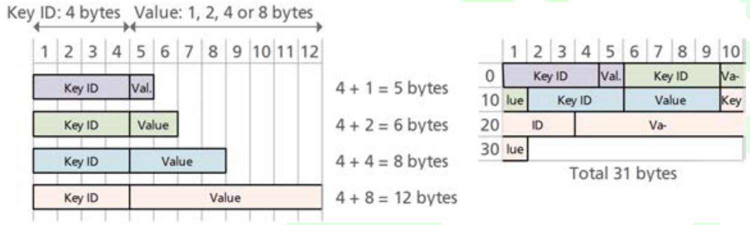
Some Configuration Items are available in the OTP Layer. They can be set or changed by fusing the eFuse.

See also OTP Layer Configuration in the Receiver Description chapter of this document.

7.5 Configuration Data

Configuration data is the binary representation of a list of Key ID and Value pairs. It is formed by concatenating keys (U4 values) and values (variable type) without any padding. This format is used in the `UBX-CFG-VALSET` and `UBX-CFG-VALGET` messages.

The figure below shows an example. The four Items (Key ID - Value pairs) on the left use the four fundamental storage sizes: one byte (L, U1, I1, E1 and X1 types), 2 bytes (U2, I2, E2 and X2 types), four byte (U4, I4, E4, X4 and R4 types) and eight bytes (U8, I8, X8 and R8 types). When concatenated (right) the Key IDs and Values are not aligned and there is no padding.



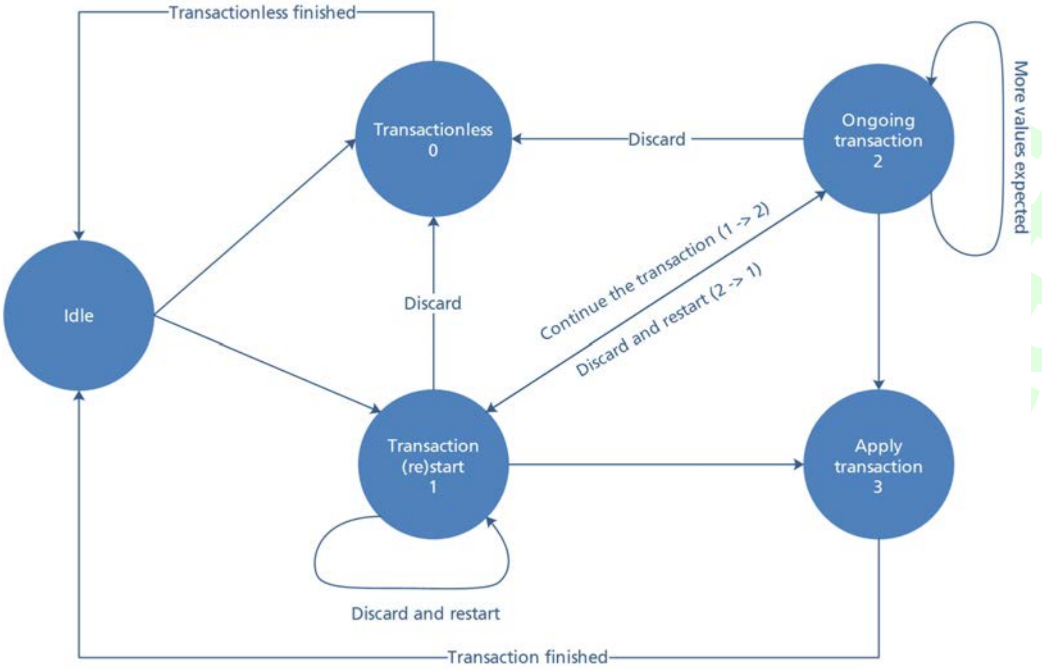
Note that this is an arbitrary example and any number of items of any value storage size can be concatenated the same way.

7.6 Configuration Transactions

The configuration concept supports two mechanisms of configuration, a transactionless mechanism where sent configuration changes are applied immediately to the configuration layer(s) requested. The second mechanism is a configuration transaction.

A transaction offers a way of queuing multiple configuration changes. It is particularly useful where different configuration keys depend on each other in such a way that sending one before the other can cause the configuration to be rejected. The queued configuration change requests are stored then checked collectively before being applied to the receiver.

A transaction can have the following states described in the figure below.



When starting a transaction, the user must specify the layer(s) the changes will be applied to. This list of configuration layer(s) must be observed throughout the transaction states, modifying the configuration layer(s) mid-transaction will cause the transaction to be aborted and no queued changes will be applied.

In the start transaction state, the receiver will lock the configuration database so that changes from another entity or message cannot be applied. It is possible to send a configuration key-value pairs with the start transaction state, and that will be queued waiting to be applied.

In the ongoing state, a configuration key and value must be sent, the receiver will abort the transaction and not apply any changes if this condition is violated, key-value pairs sent in the ongoing state will be queued waiting to be applied.

In the apply state, the queued changes will be collectively checked and applied to the requested configuration layer(s). Note that any additional key-value pairs sent within the apply state will be ignored.

Note that a transaction can only come from a single source, a `UBX-CFG-VALSET` message or a `UBX-CFG-VALDEL` message. This means that in any given transaction it is not possible to mix a delete and a save request, starting a transaction from a different source will abort the current transaction and no queued changes would be applied.

Please refer to `UBX-CFG-VALSET` and `UBX-CFG-VALDEL` messages for a detailed description of how to setup a configuration transaction, its limitations and conditions that would cause the transaction to be rejected.

7.7 Reset Behaviour

The RAM layer is always rebuilt from the layers below when the chip's processor comes out from reset. When using `UBX-CFG-RST` the processor goes through a reset cycle with these reset types (*resetMode* field):

- `0x00` hardware reset (watchdog) immediately
- `0x01` controlled software reset
- `0x04` hardware reset (watchdog) after shutdown

7.8 Configuration Reference

See Configuration Defaults for the default values.

7.8.1 CFG-CLOCK: System Clock Configuration

Configuration of system clock tree.

CFG-CLOCK-* Configuration Items

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Description*
CFG-CLOCK-OSC_FREQ	`0x40a4000d`	U4	-	Hz	Oscillator speed

7.8.2 CFG-GEOFENCE: Geofencing Configuration

See the Geofencing description for feature details.

If the receiver is sent a valid new configuration, it will respond with a `UBX-ACK-ACK` message and immediately change to the new configuration. Otherwise the receiver will reject the request, by issuing a `UBX-ACK-NAK` and continuing operation with the previous configuration.

Note that the acknowledge message does not indicate whether the PIO configuration has been successfully applied (pin assigned), it only indicates the successful configuration of the feature. The configured PIO must be previously unoccupied for successful assignment.

CFG-GEOFENCE-* Configuration Items

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Description*
CFG-GEOFENCE-CONFLVL	`0x20240011`	E1	-	-	Required confidence level for state evaluation
This value times the position's standard deviation (sigma) defines the confidence band. See Constants for CFG-GEOFENCE-CONFLVL below for a list of possible constants for this item.					
CFG-GEOFENCE-USE_PIO	`0x10240012`	L	-	-	Use PIO combined fence state output
CFG-GEOFENCE-PINPOL	`0x20240013`	E1	-	-	PIO pin polarity
See Constants for CFG-GEOFENCE-PINPOL below for a list of possible constants for this item.					
CFG-GEOFENCE-PIN	`0x20240014`	U1	-	-	PIO pin number
CFG-GEOFENCE-USE_FENCE1	`0x10240020`	L	-	-	Use first geofence
CFG-GEOFENCE-FENCE1_LAT	`0x40240021`	I4	1e-7	deg	Latitude of the first geofence circle center
CFG-GEOFENCE-FENCE1_LON	`0x40240022`	I4	1e-7	deg	Longitude of the first geofence circle center
CFG-GEOFENCE-FENCE1_RAD	`0x40240023`	U4	0.01	m	Radius of the first geofence circle
CFG-GEOFENCE-USE_FENCE2	`0x10240030`	L	-	-	Use second geofence
CFG-GEOFENCE-FENCE2_LAT	`0x40240031`	I4	1e-7	deg	Latitude of the second geofence circle center
CFG-GEOFENCE-FENCE2_LON	`0x40240032`	I4	1e-7	deg	Longitude of the second geofence circle center
CFG-GEOFENCE-FENCE2_RAD	`0x40240033`	U4	0.01	m	Radius of the second geofence circle
CFG-GEOFENCE-USE_FENCE3	`0x10240040`	L	-	-	Use third geofence
CFG-GEOFENCE-FENCE3_LAT	`0x40240041`	I4	1e-7	deg	Latitude of the third geofence circle center
CFG-GEOFENCE-FENCE3_LON	`0x40240042`	I4	1e-7	deg	Longitude of the third geofence circle center
CFG-GEOFENCE-FENCE3_RAD	`0x40240043`	U4	0.01	m	Radius of the third geofence circle

CFG-GEOFENCE- Configuration Items continued*

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Description*
CFG-GEOFENCE-USE_FENCE4	`0x10240050`	L	-	-	Use fourth geofence
CFG-GEOFENCE-FENCE4_LAT	`0x40240051`	I4	1e-7	deg	Latitude of the fourth geofence circle center
CFG-GEOFENCE-FENCE4_LON	`0x40240052`	I4	1e-7	deg	Longitude of the fourth geofence circle center
CFG-GEOFENCE-FENCE4_RAD	`0x40240053`	U4	0.01	m	Radius of the fourth geofence circle

Constants for CFG-GEOFENCE-CONFLVL

Constant	*Value*	*Description*
L000	`0`	No confidence
L680	`1`	68%
L950	`2`	95%
L997	`3`	99.7%

Constants for CFG-GEOFENCE-PINPOL

Constant	*Value*	*Description*
LOW_IN	`0`	PIO low means inside geofence
LOW_OUT	`1`	PIO low means outside geofence

7.8.3 CFG-HW: Hardware Configuration

Hardware configuration settings.

CFG-HW-* Configuration Items

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Description*
CFG-HW-DCDC_DIS	`0x10a30018`	L	-	-	DCDC converter disabled
If set to true, DCDC converter is disabled (default), otherwise DCDC converter is enabled.					
CFG-HW-SINGLE_CLK	`0x10a30019`	L	-	-	Single clock system
If set to true (default), it indicates RTC clock is present and used, otherwise main oscillator is used.					
CFG-HW-OSC_TYPE	`0x20a30025`	E1	-	-	Oscillator type
See Constants for CFG-HW-OSC_TYPE below for a list of possible constants for this item.					
CFG-HW-CLK_OFFSET	`0x40a30028`	I4	-	ppb	Clock offset
CFG-HW-CLK_OFFSET_VALID	`0x10a30029`	L	-	-	Clock offset valid
CFG-HW-CLK_PRECISION	`0x40a3002a`	U4	-	ppb	Precision of the clock offset
CFG-HW-CLK_MAX_CALIB_DEV	`0x40a3002b`	U4	-	ppb	Maximum calibration deviation
CFG-HW-CLK_MAX_CALIB_DEV_VALID	`0x10a3002c`	L	-	-	Max calibration deviation valid
CFG-HW-ANT_CFG_VOLTCTRL	`0x10a3002e`	L	-	-	Active antenna voltage control flag
Enable active antenna voltage control flag.					
CFG-HW-ANT_CFG_SHORTDET	`0x10a3002f`	L	-	-	Short antenna detection flag
Enable short antenna detection flag.					
CFG-HW-ANT_CFG_OPENDET	`0x10a30031`	L	-	-	Open antenna detection flag
Enable open antenna detection flag.					
CFG-HW-ANT_CFG_PWRDOWN	`0x10a30033`	L	-	-	Power down antenna flag
Enable power down antenna logic in the event of antenna short circuit. CFG-HW-ANT_CFG_SHORTDET must be enabled to use this feature.					

CFG-HW- Configuration Items continued*

Configuration Item	Key ID	Type	Scale	Unit	Description
CFG-HW-ANT_CFG_RECOVER	0x10a30035	L	-	-	Automatic recovery from short state flag
Enable automatic recovery from short state.					
CFG-HW-ANT_SUP_SWITCH_PIN	0x20a30036	U1	-	-	ANT1 PIO number
Antenna Switch (ANT1) PIO number.					
CFG-HW-ANT_SUP_SHORT_PIN	0x20a30037	U1	-	-	ANT0 PIO number
Antenna Short (ANT0) PIO number.					
CFG-HW-ANT_SUP_OPEN_PIN	0x20a30038	U1	-	-	ANT2 PIO number
Antenna Switch (ANT2) PIO number.					
CFG-HW-RFDC_TIMEOUT	0x20a30050	U1	-	s	RF DC Failure Timeout Setting
If the RF DC power fails, the system reset behavior is dictated by this value: 0: system shall reset immediately after a power failure is detected (default) 1 - 254: system shall reset after this many seconds, unless the RF DC power returns 255: system shall never reset after the RF DC power fails					

Constants for CFG-HW-OSC_TYPE

Constant	Value	Description
TCXO_D1V2	0x00	TCXO, direct supply 1.2V
TCXO_D1V25	0x01	TCXO, direct supply 1.25V
TCXO_D1V5	0x02	TCXO, direct supply 1.5V
TCXO_D1V8	0x03	TCXO, direct supply 1.8V
TCXO105_D1V2	0x04	TCXO 105'C, direct supply 1.2V
TCXO105_D1V25	0x05	TCXO 105'C, direct supply 1.25V
TCXO105_D1V5	0x06	TCXO 105'C, direct supply 1.5V
TCXO105_D1V8	0x07	TCXO 105'C, direct supply 1.8V
XTO_19PF_AUTO	0x08	XTO with autotuning 19pF
XTO_7PF_AUTO	0x09	XTO with autotuning 7pF
XTO_19PF	0x0a	XTO with 19pF
XTO_7PF	0x0b	XTO with 7pF

7.8.4 CFG-I2C: Configuration of the I2C Interface

Settings needed to configure the I2C communication interface.

CFG-I2C-* Configuration Items

Configuration Item	Key ID	Type	Scale	Unit	Description
CFG-I2C-ADDRESS	0x20510001	U1	-	-	I2C slave address of the receiver
CFG-I2C-EXTENDEDTIMEOUT	0x10510002	L	-	-	Flag to disable timeouting the interface after 1.5 s
CFG-I2C-ENABLED	0x10510003	L	-	-	Flag to indicate if the I2C interface should be enabled

7.8.5 CFG-I2CINPROT: Input Protocol Configuration of the I2C Interface

Input protocol enable flags of the I2C interface.

CFG-I2CINPROT-* Configuration Items

Configuration Item	Key ID	Type	Scale	Unit	Description

CFG-I2CINPROT- Configuration Items continued*

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Description*
CFG-I2CINPROT-UBX	`0x10710001`	L	-	-	Flag to indicate if UBX should be an input protocol on I2C
CFG-I2CINPROT-NMEA	`0x10710002`	L	-	-	Flag to indicate if NMEA should be an input protocol on I2C
CFG-I2CINPROT-RTCM3X	`0x10710004`	L	-	-	Flag to indicate if RTCM3X should be an input protocol on I2C

7.8.6 CFG-I2COUTPROT: Output Protocol Configuration of the I2C Interface

Output protocol enable flags of the I2C interface.

CFG-I2COUTPROT-* Configuration Items

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Description*
CFG-I2COUTPROT-UBX	`0x10720001`	L	-	-	Flag to indicate if UBX should be an output protocol on I2C
CFG-I2COUTPROT-NMEA	`0x10720002`	L	-	-	Flag to indicate if NMEA should be an output protocol on I2C
CFG-I2COUTPROT-RTCM3X	`0x10720004`	L	-	-	Flag to indicate if RTCM3X should be an output protocol on I2C

7.8.7 CFG-INFMSG: Inf Message Configuration

Information message configuration for the NMEA and UBX protocols.

CFG-INFMSG-* Configuration Items

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Description*
CFG-INFMSG-UBX_I2C	`0x20920001`	X1	-	-	Information message enable flags for the UBX protocol on the I2C interface
See Constants for CFG-INFMSG-UBX_I2C below for a list of possible constants for this item.					
CFG-INFMSG-UBX_UART1	`0x20920002`	X1	-	-	Information message enable flags for the UBX protocol on the UART1 interface
See Constants for CFG-INFMSG-UBX_UART1 below for a list of possible constants for this item.					
CFG-INFMSG-UBX_UART2	`0x20920003`	X1	-	-	Information message enable flags for the UBX protocol on the UART2 interface
See Constants for CFG-INFMSG-UBX_UART2 below for a list of possible constants for this item.					
CFG-INFMSG-UBX_USB	`0x20920004`	X1	-	-	Information message enable flags for the UBX protocol on the USB interface
See Constants for CFG-INFMSG-UBX_USB below for a list of possible constants for this item.					
CFG-INFMSG-UBX_SPI	`0x20920005`	X1	-	-	Information message enable flags for the UBX protocol on the SPI interface
See Constants for CFG-INFMSG-UBX_SPI below for a list of possible constants for this item.					
CFG-INFMSG-NMEA_I2C	`0x20920006`	X1	-	-	Information message enable flags for the NMEA protocol on the I2C interface
See Constants for CFG-INFMSG-NMEA_I2C below for a list of possible constants for this item.					

CFG-INFMSG- Configuration Items continued*

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Description*
CFG-INFMSG-NMEA_UART1	`0x20920007`	X1	-	-	Information message enable flags for the NMEA protocol on the UART1 interface
See Constants for CFG-INFMSG-NMEA_UART1 below for a list of possible constants for this item.					
CFG-INFMSG-NMEA_UART2	`0x20920008`	X1	-	-	Information message enable flags for the NMEA protocol on the UART2 interface
See Constants for CFG-INFMSG-NMEA_UART2 below for a list of possible constants for this item.					
CFG-INFMSG-NMEA_USB	`0x20920009`	X1	-	-	Information message enable flags for the NMEA protocol on the USB interface
See Constants for CFG-INFMSG-NMEA_USB below for a list of possible constants for this item.					
CFG-INFMSG-NMEA_SPI	`0x2092000a`	X1	-	-	Information message enable flags for the NMEA protocol on the SPI interface
See Constants for CFG-INFMSG-NMEA_SPI below for a list of possible constants for this item.					

Constants for CFG-INFMSG-UBX_I2C

Constant	*Value*	*Description*
ERROR	`0x01`	Enable ERROR information messages
WARNING	`0x02`	Enable WARNING information messages
NOTICE	`0x04`	Enable NOTICE information messages
TEST	`0x08`	Enable TEST information messages
DEBUG	`0x10`	Enable DEBUG information messages

Constants for CFG-INFMSG-UBX_UART1

Constant	*Value*	*Description*
ERROR	`0x01`	Enable ERROR information messages
WARNING	`0x02`	Enable WARNING information messages
NOTICE	`0x04`	Enable NOTICE information messages
TEST	`0x08`	Enable TEST information messages
DEBUG	`0x10`	Enable DEBUG information messages

Constants for CFG-INFMSG-UBX_UART2

Constant	*Value*	*Description*
ERROR	`0x01`	Enable ERROR information messages
WARNING	`0x02`	Enable WARNING information messages
NOTICE	`0x04`	Enable NOTICE information messages
TEST	`0x08`	Enable TEST information messages
DEBUG	`0x10`	Enable DEBUG information messages

Constants for CFG-INFMSG-UBX_USB

Constant	*Value*	*Description*
ERROR	`0x01`	Enable ERROR information messages
WARNING	`0x02`	Enable WARNING information messages
NOTICE	`0x04`	Enable NOTICE information messages

Constants for CFG-INFMSG-UBX_USB continued

Constant	Value	Description
TEST	0x08	Enable TEST information messages
DEBUG	0x10	Enable DEBUG information messages

Constants for CFG-INFMSG-UBX_SPI

Constant	Value	Description
ERROR	0x01	Enable ERROR information messages
WARNING	0x02	Enable WARNING information messages
NOTICE	0x04	Enable NOTICE information messages
TEST	0x08	Enable TEST information messages
DEBUG	0x10	Enable DEBUG information messages

Constants for CFG-INFMSG-NMEA_I2C

Constant	Value	Description
ERROR	0x01	Enable ERROR information messages
WARNING	0x02	Enable WARNING information messages
NOTICE	0x04	Enable NOTICE information messages
TEST	0x08	Enable TEST information messages
DEBUG	0x10	Enable DEBUG information messages

Constants for CFG-INFMSG-NMEA_UART1

Constant	Value	Description
ERROR	0x01	Enable ERROR information messages
WARNING	0x02	Enable WARNING information messages
NOTICE	0x04	Enable NOTICE information messages
TEST	0x08	Enable TEST information messages
DEBUG	0x10	Enable DEBUG information messages

Constants for CFG-INFMSG-NMEA_UART2

Constant	Value	Description
ERROR	0x01	Enable ERROR information messages
WARNING	0x02	Enable WARNING information messages
NOTICE	0x04	Enable NOTICE information messages
TEST	0x08	Enable TEST information messages
DEBUG	0x10	Enable DEBUG information messages

Constants for CFG-INFMSG-NMEA_USB

Constant	Value	Description
ERROR	0x01	Enable ERROR information messages
WARNING	0x02	Enable WARNING information messages
NOTICE	0x04	Enable NOTICE information messages
TEST	0x08	Enable TEST information messages
DEBUG	0x10	Enable DEBUG information messages

Constants for CFG-INFMSG-NMEA_SPI

Constant	Value	Description
ERROR	0x01	Enable ERROR information messages

Constants for CFG-INFMSG-NMEA_SPI continued

Constant	*Value*	*Description*
WARNING	`0x02`	Enable WARNING information messages
NOTICE	`0x04`	Enable NOTICE information messages
TEST	`0x08`	Enable TEST information messages
DEBUG	`0x10`	Enable DEBUG information messages

7.8.8 CFG-ITFM: Jamming/Interference Monitor configuration

Configuration of Jamming/Interference monitor.

CFG-ITFM-* Configuration Items

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Description*
CFG-ITFM-BBTHRESHOLD	`0x20410001`	U1	-	-	Broadband jamming detection threshold
CFG-ITFM-CWTHRESHOLD	`0x20410002`	U1	-	-	CW jamming detection threshold
CFG-ITFM-ENABLE	`0x1041000d`	L	-	-	Enable interference detection
CFG-ITFM-ANTSETTING	`0x20410010`	E1	-	-	Antenna setting
See Constants for CFG-ITFM-ANTSETTING below for a list of possible constants for this item.					
CFG-ITFM-ENABLE_AUX	`0x10410013`	L	-	-	Set to true to scan auxiliary bands
Supported on u-blox 8 / u-blox M8 only, otherwise ignored.					

Constants for CFG-ITFM-ANTSETTING

Constant	*Value*	*Description*
UNKNOWN	`0`	Unknown
PASSIVE	`1`	Passive
ACTIVE	`2`	Active

7.8.9 CFG-LOGFILTER: Data Logger Configuration

This group can be used to configure the data logger, i.e. to enable/disable the log recording and to get/set the position entry filter settings.

Position entries can be filtered based on time difference, position difference or current speed thresholds. Position and speed filtering also have a minimum time interval. A position is logged if any of the thresholds are exceeded. If a threshold is set to zero it is ignored. The maximum rate of position logging is 1Hz.

The filter settings will be configured to the provided values only if the 'applyAllFilterSettings' flag is set. This allows the recording to be enabled/disabled independently of configuring the filter settings.

It is possible to configure the data logger in the absence of a logging file. By doing so, once the logging file is created, the data logger configuration will take effect immediately and logging recording and filtering will activate according to the configuration.

CFG-LOGFILTER-* Configuration Items

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Description*
CFG-LOGFILTER-RECORD_ENA	`0x10de0002`	L	-	-	Recording enabled
Set to true when recording enabled.					
CFG-LOGFILTER-ONCE_PER_WAKE_UP_ENA	`0x10de0003`	L	-	-	Once per wakeup
Set to true recording only one single position per PSM on/off mode wake-up period is enabled. Note: the value set here does not take effect unless CFG-LOGFILTER-APPLY_ALL_FILTERS is enabled.					

CFG-LOGFILTER- Configuration Items continued*

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Description*
CFG-LOGFILTER-APPLY_ALL_FILTERS	`0x10de0004`	L	-	-	Apply all filter settings
Set to true when all filter settings are to be applied, not just recording enabling/disabling.					
CFG-LOGFILTER-MIN_INTERVAL	`0x30de0005`	U2	-	s	Minimum time interval between logged positions
Minimum time interval between logged positions (0 = not set). **This is only applied in combination with the speed and/or position thresholds.** If both MIN_INTERVAL and TIME_THRS are set, MIN_INTERVAL must be less than or equal to TIME_THRS. Note: the value set here does not take effect unless CFG-LOGFILTER-APPLY_ALL_FILTERS is enabled.					
CFG-LOGFILTER-TIME_THRS	`0x30de0006`	U2	-	s	Time threshold
If the time difference is greater than the threshold then the position is logged (0 = not set). Note: the value set here does not take effect unless CFG-LOGFILTER-APPLY_ALL_FILTERS is enabled.					
CFG-LOGFILTER-SPEED_THRS	`0x30de0007`	U2	-	m/s	Speed threshold
If the current speed is greater than the threshold then the position is logged (0 = not set). MIN_INTERVAL also applies. Note: value set here does not take effect unless CFG-LOGFILTER-APPLY_ALL_FILTERS is enabled.					
CFG-LOGFILTER-POSITION_THRS	`0x40de0008`	U4	-	m	Position threshold
If the 3D position difference is greater than the threshold then the position is logged (0 = not set). MIN_INTERVAL also applies. Note: the value set here does not take effect unless CFG-LOGFILTER-APPLY_ALL_FILTERS is enabled.					

7.8.10 CFG-MOT: Motion Detector Configuration

The items in this group specify the parameters used for the internal receiver motion detector. The platform motion is assessed by combining the detected motion of different detectors looking at specific data types (i.e. GNSS, gyroscopes, accelerometers, wheel-ticks). The decision thresholds of the internal detectors can be specified using the configuration items in this group.

CFG-MOT-* Configuration Items

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Description*
CFG-MOT-GNSSSPEED_THRS	`0x20250038`	U1	0.01	m/s	GNSS speed threshold below which platform is considered as stationary (a.k.a. static hold threshold)
Set this paramter to 0 for a firmware default value or bahaviour.					
CFG-MOT-GNSSDIST_THRS	`0x3025003b`	U2	-	-	Distance above which GNSS-based stationary motion is exit (a.k.a. static hold distance threshold)
Set this paramter to 0 for a firmware default value or bahaviour.					

7.8.11 CFG-MSGOUT: Message Output Configuration

For each message and port a separate output rate (per second, per epoch) can be configured.

CFG-MSGOUT-* Configuration Items

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Description*
CFG-MSGOUT-NMEA_ID_DTM_I2C	`0x209100a6`	U1	-	-	Output rate of the NMEA-GX-DTM message on port I2C

CFG-MSGOUT-* Configuration Items continued

Configuration Item	Key ID	Type	Scale	Unit	Description
CFG-MSGOUT-NMEA_ID_DTM_SPI	0x209100aa	U1	-	-	Output rate of the NMEA-GX-DTM message on port SPI
CFG-MSGOUT-NMEA_ID_DTM_UART1	0x209100a7	U1	-	-	Output rate of the NMEA-GX-DTM message on port UART1
CFG-MSGOUT-NMEA_ID_DTM_UART2	0x209100a8	U1	-	-	Output rate of the NMEA-GX-DTM message on port UART2
CFG-MSGOUT-NMEA_ID_DTM_USB	0x209100a9	U1	-	-	Output rate of the NMEA-GX-DTM message on port USB
CFG-MSGOUT-NMEA_ID_GBS_I2C	0x209100dd	U1	-	-	Output rate of the NMEA-GX-GBS message on port I2C
CFG-MSGOUT-NMEA_ID_GBS_SPI	0x209100e1	U1	-	-	Output rate of the NMEA-GX-GBS message on port SPI
CFG-MSGOUT-NMEA_ID_GBS_UART1	0x209100de	U1	-	-	Output rate of the NMEA-GX-GBS message on port UART1
CFG-MSGOUT-NMEA_ID_GBS_UART2	0x209100df	U1	-	-	Output rate of the NMEA-GX-GBS message on port UART2
CFG-MSGOUT-NMEA_ID_GBS_USB	0x209100e0	U1	-	-	Output rate of the NMEA-GX-GBS message on port USB
CFG-MSGOUT-NMEA_ID_GGA_I2C	0x209100ba	U1	-	-	Output rate of the NMEA-GX-GGA message on port I2C
CFG-MSGOUT-NMEA_ID_GGA_SPI	0x209100be	U1	-	-	Output rate of the NMEA-GX-GGA message on port SPI
CFG-MSGOUT-NMEA_ID_GGA_UART1	0x209100bb	U1	-	-	Output rate of the NMEA-GX-GGA message on port UART1
CFG-MSGOUT-NMEA_ID_GGA_UART2	0x209100bc	U1	-	-	Output rate of the NMEA-GX-GGA message on port UART2
CFG-MSGOUT-NMEA_ID_GGA_USB	0x209100bd	U1	-	-	Output rate of the NMEA-GX-GGA message on port USB
CFG-MSGOUT-NMEA_ID_GLL_I2C	0x209100c9	U1	-	-	Output rate of the NMEA-GX-GLL message on port I2C
CFG-MSGOUT-NMEA_ID_GLL_SPI	0x209100cd	U1	-	-	Output rate of the NMEA-GX-GLL message on port SPI
CFG-MSGOUT-NMEA_ID_GLL_UART1	0x209100ca	U1	-	-	Output rate of the NMEA-GX-GLL message on port UART1
CFG-MSGOUT-NMEA_ID_GLL_UART2	0x209100cb	U1	-	-	Output rate of the NMEA-GX-GLL message on port UART2
CFG-MSGOUT-NMEA_ID_GLL_USB	0x209100cc	U1	-	-	Output rate of the NMEA-GX-GLL message on port USB
CFG-MSGOUT-NMEA_ID_GNS_I2C	0x209100b5	U1	-	-	Output rate of the NMEA-GX-GNS message on port I2C
CFG-MSGOUT-NMEA_ID_GNS_SPI	0x209100b9	U1	-	-	Output rate of the NMEA-GX-GNS message on port SPI
CFG-MSGOUT-NMEA_ID_GNS_UART1	0x209100b6	U1	-	-	Output rate of the NMEA-GX-GNS message on port UART1

CFG-MSGOUT-* Configuration Items continued

Configuration Item	Key ID	Type	Scale	Unit	Description
CFG-MSGOUT-NMEA_ID_GNS_UART2	0x209100b7	U1	-	-	Output rate of the NMEA-GX-GNS message on port UART2
CFG-MSGOUT-NMEA_ID_GNS_USB	0x209100b8	U1	-	-	Output rate of the NMEA-GX-GNS message on port USB
CFG-MSGOUT-NMEA_ID_GRS_I2C	0x209100ce	U1	-	-	Output rate of the NMEA-GX-GRS message on port I2C
CFG-MSGOUT-NMEA_ID_GRS_SPI	0x209100d2	U1	-	-	Output rate of the NMEA-GX-GRS message on port SPI
CFG-MSGOUT-NMEA_ID_GRS_UART1	0x209100cf	U1	-	-	Output rate of the NMEA-GX-GRS message on port UART1
CFG-MSGOUT-NMEA_ID_GRS_UART2	0x209100d0	U1	-	-	Output rate of the NMEA-GX-GRS message on port UART2
CFG-MSGOUT-NMEA_ID_GRS_USB	0x209100d1	U1	-	-	Output rate of the NMEA-GX-GRS message on port USB
CFG-MSGOUT-NMEA_ID_GSA_I2C	0x209100bf	U1	-	-	Output rate of the NMEA-GX-GSA message on port I2C
CFG-MSGOUT-NMEA_ID_GSA_SPI	0x209100c3	U1	-	-	Output rate of the NMEA-GX-GSA message on port SPI
CFG-MSGOUT-NMEA_ID_GSA_UART1	0x209100c0	U1	-	-	Output rate of the NMEA-GX-GSA message on port UART1
CFG-MSGOUT-NMEA_ID_GSA_UART2	0x209100c1	U1	-	-	Output rate of the NMEA-GX-GSA message on port UART2
CFG-MSGOUT-NMEA_ID_GSA_USB	0x209100c2	U1	-	-	Output rate of the NMEA-GX-GSA message on port USB
CFG-MSGOUT-NMEA_ID_GST_I2C	0x209100d3	U1	-	-	Output rate of the NMEA-GX-GST message on port I2C
CFG-MSGOUT-NMEA_ID_GST_SPI	0x209100d7	U1	-	-	Output rate of the NMEA-GX-GST message on port SPI
CFG-MSGOUT-NMEA_ID_GST_UART1	0x209100d4	U1	-	-	Output rate of the NMEA-GX-GST message on port UART1
CFG-MSGOUT-NMEA_ID_GST_UART2	0x209100d5	U1	-	-	Output rate of the NMEA-GX-GST message on port UART2
CFG-MSGOUT-NMEA_ID_GST_USB	0x209100d6	U1	-	-	Output rate of the NMEA-GX-GST message on port USB
CFG-MSGOUT-NMEA_ID_GSV_I2C	0x209100c4	U1	-	-	Output rate of the NMEA-GX-GSV message on port I2C
CFG-MSGOUT-NMEA_ID_GSV_SPI	0x209100c8	U1	-	-	Output rate of the NMEA-GX-GSV message on port SPI
CFG-MSGOUT-NMEA_ID_GSV_UART1	0x209100c5	U1	-	-	Output rate of the NMEA-GX-GSV message on port UART1
CFG-MSGOUT-NMEA_ID_GSV_UART2	0x209100c6	U1	-	-	Output rate of the NMEA-GX-GSV message on port UART2
CFG-MSGOUT-NMEA_ID_GSV_USB	0x209100c7	U1	-	-	Output rate of the NMEA-GX-GSV message on port USB

CFG-MSGOUT- Configuration Items continued*

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Description*
CFG-MSGOUT-NMEA_ID_RMC_I2C	`0x209100ab`	U1	-	-	Output rate of the NMEA-GX-RMC message on port I2C
CFG-MSGOUT-NMEA_ID_RMC_SPI	`0x209100af`	U1	-	-	Output rate of the NMEA-GX-RMC message on port SPI
CFG-MSGOUT-NMEA_ID_RMC_UART1	`0x209100ac`	U1	-	-	Output rate of the NMEA-GX-RMC message on port UART1
CFG-MSGOUT-NMEA_ID_RMC_UART2	`0x209100ad`	U1	-	-	Output rate of the NMEA-GX-RMC message on port UART2
CFG-MSGOUT-NMEA_ID_RMC_USB	`0x209100ae`	U1	-	-	Output rate of the NMEA-GX-RMC message on port USB
CFG-MSGOUT-NMEA_ID_VLW_I2C	`0x209100e7`	U1	-	-	Output rate of the NMEA-GX-VLW message on port I2C
CFG-MSGOUT-NMEA_ID_VLW_SPI	`0x209100eb`	U1	-	-	Output rate of the NMEA-GX-VLW message on port SPI
CFG-MSGOUT-NMEA_ID_VLW_UART1	`0x209100e8`	U1	-	-	Output rate of the NMEA-GX-VLW message on port UART1
CFG-MSGOUT-NMEA_ID_VLW_UART2	`0x209100e9`	U1	-	-	Output rate of the NMEA-GX-VLW message on port UART2
CFG-MSGOUT-NMEA_ID_VLW_USB	`0x209100ea`	U1	-	-	Output rate of the NMEA-GX-VLW message on port USB
CFG-MSGOUT-NMEA_ID_VTG_I2C	`0x209100b0`	U1	-	-	Output rate of the NMEA-GX-VTG message on port I2C
CFG-MSGOUT-NMEA_ID_VTG_SPI	`0x209100b4`	U1	-	-	Output rate of the NMEA-GX-VTG message on port SPI
CFG-MSGOUT-NMEA_ID_VTG_UART1	`0x209100b1`	U1	-	-	Output rate of the NMEA-GX-VTG message on port UART1
CFG-MSGOUT-NMEA_ID_VTG_UART2	`0x209100b2`	U1	-	-	Output rate of the NMEA-GX-VTG message on port UART2
CFG-MSGOUT-NMEA_ID_VTG_USB	`0x209100b3`	U1	-	-	Output rate of the NMEA-GX-VTG message on port USB
CFG-MSGOUT-NMEA_ID_ZDA_I2C	`0x209100d8`	U1	-	-	Output rate of the NMEA-GX-ZDA message on port I2C
CFG-MSGOUT-NMEA_ID_ZDA_SPI	`0x209100dc`	U1	-	-	Output rate of the NMEA-GX-ZDA message on port SPI
CFG-MSGOUT-NMEA_ID_ZDA_UART1	`0x209100d9`	U1	-	-	Output rate of the NMEA-GX-ZDA message on port UART1
CFG-MSGOUT-NMEA_ID_ZDA_UART2	`0x209100da`	U1	-	-	Output rate of the NMEA-GX-ZDA message on port UART2
CFG-MSGOUT-NMEA_ID_ZDA_USB	`0x209100db`	U1	-	-	Output rate of the NMEA-GX-ZDA message on port USB
CFG-MSGOUT-PUBX_ID_POLYP_I2C	`0x209100ec`	U1	-	-	Output rate of the NMEA-GX-PUBX00 message on port I2C
CFG-MSGOUT-PUBX_ID_POLYP_SPI	`0x209100f0`	U1	-	-	Output rate of the NMEA-GX-PUBX00 message on port SPI

CFG-MSGOUT- Configuration Items continued*

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Description*
CFG-MSGOUT-PUBX_ID_POLYP_UART1	`0x209100ed`	U1	-	-	Output rate of the NMEA-GX-PUBX00 message on port UART1
CFG-MSGOUT-PUBX_ID_POLYP_UART2	`0x209100ee`	U1	-	-	Output rate of the NMEA-GX-PUBX00 message on port UART2
CFG-MSGOUT-PUBX_ID_POLYP_USB	`0x209100ef`	U1	-	-	Output rate of the NMEA-GX-PUBX00 message on port USB
CFG-MSGOUT-PUBX_ID_POLYS_I2C	`0x209100f1`	U1	-	-	Output rate of the NMEA-GX-PUBX03 message on port I2C
CFG-MSGOUT-PUBX_ID_POLYS_SPI	`0x209100f5`	U1	-	-	Output rate of the NMEA-GX-PUBX03 message on port SPI
CFG-MSGOUT-PUBX_ID_POLYS_UART1	`0x209100f2`	U1	-	-	Output rate of the NMEA-GX-PUBX03 message on port UART1
CFG-MSGOUT-PUBX_ID_POLYS_UART2	`0x209100f3`	U1	-	-	Output rate of the NMEA-GX-PUBX03 message on port UART2
CFG-MSGOUT-PUBX_ID_POLYS_USB	`0x209100f4`	U1	-	-	Output rate of the NMEA-GX-PUBX03 message on port USB
CFG-MSGOUT-PUBX_ID_POLYT_I2C	`0x209100f6`	U1	-	-	Output rate of the NMEA-GX-PUBX04 message on port I2C
CFG-MSGOUT-PUBX_ID_POLYT_SPI	`0x209100fa`	U1	-	-	Output rate of the NMEA-GX-PUBX04 message on port SPI
CFG-MSGOUT-PUBX_ID_POLYT_UART1	`0x209100f7`	U1	-	-	Output rate of the NMEA-GX-PUBX04 message on port UART1
CFG-MSGOUT-PUBX_ID_POLYT_UART2	`0x209100f8`	U1	-	-	Output rate of the NMEA-GX-PUBX04 message on port UART2
CFG-MSGOUT-PUBX_ID_POLYT_USB	`0x209100f9`	U1	-	-	Output rate of the NMEA-GX-PUBX04 message on port USB
CFG-MSGOUT-RTCM_3X_TYPE1005_I2C	`0x209102bd`	U1	-	-	Output rate of the RTCM-3X-TYPE1005 message on port I2C
CFG-MSGOUT-RTCM_3X_TYPE1005_SPI	`0x209102c1`	U1	-	-	Output rate of the RTCM-3X-TYPE1005 message on port SPI
CFG-MSGOUT-RTCM_3X_TYPE1005_UART1	`0x209102be`	U1	-	-	Output rate of the RTCM-3X-TYPE1005 message on port UART1
CFG-MSGOUT-RTCM_3X_TYPE1005_UART2	`0x209102bf`	U1	-	-	Output rate of the RTCM-3X-TYPE1005 message on port UART2
CFG-MSGOUT-RTCM_3X_TYPE1005_USB	`0x209102c0`	U1	-	-	Output rate of the RTCM-3X-TYPE1005 message on port USB
CFG-MSGOUT-RTCM_3X_TYPE1077_I2C	`0x209102cc`	U1	-	-	Output rate of the RTCM-3X-TYPE1077 message on port I2C
CFG-MSGOUT-RTCM_3X_TYPE1077_SPI	`0x209102d0`	U1	-	-	Output rate of the RTCM-3X-TYPE1077 message on port SPI
CFG-MSGOUT-RTCM_3X_TYPE1077_UART1	`0x209102cd`	U1	-	-	Output rate of the RTCM-3X-TYPE1077 message on port UART1
CFG-MSGOUT-RTCM_3X_TYPE1077_UART2	`0x209102ce`	U1	-	-	Output rate of the RTCM-3X-TYPE1077 message on port UART2

CFG-MSGOUT- Configuration Items continued*

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Description*
CFG-MSGOUT-RTCM_3X_TYPE1077_USB	`0x209102cf`	U1	-	-	Output rate of the RTCM-3X-TYPE1077 message on port USB
CFG-MSGOUT-RTCM_3X_TYPE1087_I2C	`0x209102d1`	U1	-	-	Output rate of the RTCM-3X-TYPE1087 message on port I2C
CFG-MSGOUT-RTCM_3X_TYPE1087_SPI	`0x209102d5`	U1	-	-	Output rate of the RTCM-3X-TYPE1087 message on port SPI
CFG-MSGOUT-RTCM_3X_TYPE1087_UART1	`0x209102d2`	U1	-	-	Output rate of the RTCM-3X-TYPE1087 message on port UART1
CFG-MSGOUT-RTCM_3X_TYPE1087_UART2	`0x209102d3`	U1	-	-	Output rate of the RTCM-3X-TYPE1087 message on port UART2
CFG-MSGOUT-RTCM_3X_TYPE1087_USB	`0x209102d4`	U1	-	-	Output rate of the RTCM-3X-TYPE1087 message on port USB
CFG-MSGOUT-RTCM_3X_TYPE1097_I2C	`0x20910318`	U1	-	-	Output rate of the RTCM-3X-TYPE1097 message on port I2C
CFG-MSGOUT-RTCM_3X_TYPE1097_SPI	`0x2091031c`	U1	-	-	Output rate of the RTCM-3X-TYPE1097 message on port SPI
CFG-MSGOUT-RTCM_3X_TYPE1097_UART1	`0x20910319`	U1	-	-	Output rate of the RTCM-3X-TYPE1097 message on port UART1
CFG-MSGOUT-RTCM_3X_TYPE1097_UART2	`0x2091031a`	U1	-	-	Output rate of the RTCM-3X-TYPE1097 message on port UART2
CFG-MSGOUT-RTCM_3X_TYPE1097_USB	`0x2091031b`	U1	-	-	Output rate of the RTCM-3X-TYPE1097 message on port USB
CFG-MSGOUT-RTCM_3X_TYPE1127_I2C	`0x209102d6`	U1	-	-	Output rate of the RTCM-3X-TYPE1127 message on port I2C
CFG-MSGOUT-RTCM_3X_TYPE1127_SPI	`0x209102da`	U1	-	-	Output rate of the RTCM-3X-TYPE1127 message on port SPI
CFG-MSGOUT-RTCM_3X_TYPE1127_UART1	`0x209102d7`	U1	-	-	Output rate of the RTCM-3X-TYPE1127 message on port UART1
CFG-MSGOUT-RTCM_3X_TYPE1127_UART2	`0x209102d8`	U1	-	-	Output rate of the RTCM-3X-TYPE1127 message on port UART2
CFG-MSGOUT-RTCM_3X_TYPE1127_USB	`0x209102d9`	U1	-	-	Output rate of the RTCM-3X-TYPE1127 message on port USB
CFG-MSGOUT-RTCM_3X_TYPE1230_I2C	`0x20910303`	U1	-	-	Output rate of the RTCM-3X-TYPE1230 message on port I2C
CFG-MSGOUT-RTCM_3X_TYPE1230_SPI	`0x20910307`	U1	-	-	Output rate of the RTCM-3X-TYPE1230 message on port SPI
CFG-MSGOUT-RTCM_3X_TYPE1230_UART1	`0x20910304`	U1	-	-	Output rate of the RTCM-3X-TYPE1230 message on port UART1
CFG-MSGOUT-RTCM_3X_TYPE1230_UART2	`0x20910305`	U1	-	-	Output rate of the RTCM-3X-TYPE1230 message on port UART2
CFG-MSGOUT-RTCM_3X_TYPE1230_USB	`0x20910306`	U1	-	-	Output rate of the RTCM-3X-TYPE1230 message on port USB
CFG-MSGOUT-UBX_AID_ALM_I2C	`0x2091016e`	U1	-	-	Output rate of the UBX-AID-ALM message on port I2C

CFG-MSGOUT- Configuration Items continued*

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Description*
CFG-MSGOUT-UBX_AID_ALM_SPI	`0x20910172`	U1	-	-	Output rate of the UBX-AID-ALM message on port SPI
CFG-MSGOUT-UBX_AID_ALM_UART1	`0x2091016f`	U1	-	-	Output rate of the UBX-AID-ALM message on port UART1
CFG-MSGOUT-UBX_AID_ALM_UART2	`0x20910170`	U1	-	-	Output rate of the UBX-AID-ALM message on port UART2
CFG-MSGOUT-UBX_AID_ALM_USB	`0x20910171`	U1	-	-	Output rate of the UBX-AID-ALM message on port USB
CFG-MSGOUT-UBX_AID_EPH_I2C	`0x20910164`	U1	-	-	Output rate of the UBX-AID-EPH message on port I2C
CFG-MSGOUT-UBX_AID_EPH_SPI	`0x20910168`	U1	-	-	Output rate of the UBX-AID-EPH message on port SPI
CFG-MSGOUT-UBX_AID_EPH_UART1	`0x20910165`	U1	-	-	Output rate of the UBX-AID-EPH message on port UART1
CFG-MSGOUT-UBX_AID_EPH_UART2	`0x20910166`	U1	-	-	Output rate of the UBX-AID-EPH message on port UART2
CFG-MSGOUT-UBX_AID_EPH_USB	`0x20910167`	U1	-	-	Output rate of the UBX-AID-EPH message on port USB
CFG-MSGOUT-UBX_AID_INI_I2C	`0x209100fb`	U1	-	-	Output rate of the UBX-AID-INI message on port I2C
CFG-MSGOUT-UBX_AID_INI_SPI	`0x209100ff`	U1	-	-	Output rate of the UBX-AID-INI message on port SPI
CFG-MSGOUT-UBX_AID_INI_UART1	`0x209100fc`	U1	-	-	Output rate of the UBX-AID-INI message on port UART1
CFG-MSGOUT-UBX_AID_INI_UART2	`0x209100fd`	U1	-	-	Output rate of the UBX-AID-INI message on port UART2
CFG-MSGOUT-UBX_AID_INI_USB	`0x209100fe`	U1	-	-	Output rate of the UBX-AID-INI message on port USB
CFG-MSGOUT-UBX_LOG_INFO_I2C	`0x20910259`	U1	-	-	Output rate of the UBX-LOG-INFO message on port I2C
CFG-MSGOUT-UBX_LOG_INFO_SPI	`0x2091025d`	U1	-	-	Output rate of the UBX-LOG-INFO message on port SPI
CFG-MSGOUT-UBX_LOG_INFO_UART1	`0x2091025a`	U1	-	-	Output rate of the UBX-LOG-INFO message on port UART1
CFG-MSGOUT-UBX_LOG_INFO_UART2	`0x2091025b`	U1	-	-	Output rate of the UBX-LOG-INFO message on port UART2
CFG-MSGOUT-UBX_LOG_INFO_USB	`0x2091025c`	U1	-	-	Output rate of the UBX-LOG-INFO message on port USB
CFG-MSGOUT-UBX_MON_HW2_I2C	`0x209101b9`	U1	-	-	Output rate of the UBX-MON-HW2 message on port I2C
CFG-MSGOUT-UBX_MON_HW2_SPI	`0x209101bd`	U1	-	-	Output rate of the UBX-MON-HW2 message on port SPI
CFG-MSGOUT-UBX_MON_HW2_UART1	`0x209101ba`	U1	-	-	Output rate of the UBX-MON-HW2 message on port UART1

CFG-MSGOUT- Configuration Items continued*

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Description*
CFG-MSGOUT-UBX_MON_HW2_UART2	`0x209101bb`	U1	-	-	Output rate of the UBX-MON-HW2 message on port UART2
CFG-MSGOUT-UBX_MON_HW2_USB	`0x209101bc`	U1	-	-	Output rate of the UBX-MON-HW2 message on port USB
CFG-MSGOUT-UBX_MON_HW_I2C	`0x209101b4`	U1	-	-	Output rate of the UBX-MON-HW message on port I2C
CFG-MSGOUT-UBX_MON_HW_SPI	`0x209101b8`	U1	-	-	Output rate of the UBX-MON-HW message on port SPI
CFG-MSGOUT-UBX_MON_HW_UART1	`0x209101b5`	U1	-	-	Output rate of the UBX-MON-HW message on port UART1
CFG-MSGOUT-UBX_MON_HW_UART2	`0x209101b6`	U1	-	-	Output rate of the UBX-MON-HW message on port UART2
CFG-MSGOUT-UBX_MON_HW_USB	`0x209101b7`	U1	-	-	Output rate of the UBX-MON-HW message on port USB
CFG-MSGOUT-UBX_MON_IO_I2C	`0x209101a5`	U1	-	-	Output rate of the UBX-MON-IO message on port I2C
CFG-MSGOUT-UBX_MON_IO_SPI	`0x209101a9`	U1	-	-	Output rate of the UBX-MON-IO message on port SPI
CFG-MSGOUT-UBX_MON_IO_UART1	`0x209101a6`	U1	-	-	Output rate of the UBX-MON-IO message on port UART1
CFG-MSGOUT-UBX_MON_IO_UART2	`0x209101a7`	U1	-	-	Output rate of the UBX-MON-IO message on port UART2
CFG-MSGOUT-UBX_MON_IO_USB	`0x209101a8`	U1	-	-	Output rate of the UBX-MON-IO message on port USB
CFG-MSGOUT-UBX_MON_MSGPP_I2C	`0x20910196`	U1	-	-	Output rate of the UBX-MON-MSGPP message on port I2C
CFG-MSGOUT-UBX_MON_MSGPP_SPI	`0x2091019a`	U1	-	-	Output rate of the UBX-MON-MSGPP message on port SPI
CFG-MSGOUT-UBX_MON_MSGPP_UART1	`0x20910197`	U1	-	-	Output rate of the UBX-MON-MSGPP message on port UART1
CFG-MSGOUT-UBX_MON_MSGPP_UART2	`0x20910198`	U1	-	-	Output rate of the UBX-MON-MSGPP message on port UART2
CFG-MSGOUT-UBX_MON_MSGPP_USB	`0x20910199`	U1	-	-	Output rate of the UBX-MON-MSGPP message on port USB
CFG-MSGOUT-UBX_MON_PT2_I2C	`0x20910209`	U1	-	-	Output rate of the UBX-MON-PT2 message on port I2C
CFG-MSGOUT-UBX_MON_PT2_SPI	`0x2091020d`	U1	-	-	Output rate of the UBX-MON-PT2 message on port SPI
CFG-MSGOUT-UBX_MON_PT2_UART1	`0x2091020a`	U1	-	-	Output rate of the UBX-MON-PT2 message on port UART1
CFG-MSGOUT-UBX_MON_PT2_UART2	`0x2091020b`	U1	-	-	Output rate of the UBX-MON-PT2 message on port UART2
CFG-MSGOUT-UBX_MON_PT2_USB	`0x2091020c`	U1	-	-	Output rate of the UBX-MON-PT2 message on port USB

CFG-MSGOUT- Configuration Items continued*

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Description*
CFG-MSGOUT-UBX_MON_RXBUF_I2C	`0x209101a0`	U1	-	-	Output rate of the UBX-MON-RXBUF message on port I2C
CFG-MSGOUT-UBX_MON_RXBUF_SPI	`0x209101a4`	U1	-	-	Output rate of the UBX-MON-RXBUF message on port SPI
CFG-MSGOUT-UBX_MON_RXBUF_UART1	`0x209101a1`	U1	-	-	Output rate of the UBX-MON-RXBUF message on port UART1
CFG-MSGOUT-UBX_MON_RXBUF_UART2	`0x209101a2`	U1	-	-	Output rate of the UBX-MON-RXBUF message on port UART2
CFG-MSGOUT-UBX_MON_RXBUF_USB	`0x209101a3`	U1	-	-	Output rate of the UBX-MON-RXBUF message on port USB
CFG-MSGOUT-UBX_MON_RXR_I2C	`0x20910187`	U1	-	-	Output rate of the UBX-MON-RXR message on port I2C
CFG-MSGOUT-UBX_MON_RXR_SPI	`0x2091018b`	U1	-	-	Output rate of the UBX-MON-RXR message on port SPI
CFG-MSGOUT-UBX_MON_RXR_UART1	`0x20910188`	U1	-	-	Output rate of the UBX-MON-RXR message on port UART1
CFG-MSGOUT-UBX_MON_RXR_UART2	`0x20910189`	U1	-	-	Output rate of the UBX-MON-RXR message on port UART2
CFG-MSGOUT-UBX_MON_RXR_USB	`0x2091018a`	U1	-	-	Output rate of the UBX-MON-RXR message on port USB
CFG-MSGOUT-UBX_MON_TEMP_I2C	`0x20910331`	U1	-	-	Temperature measurement result and temperature related warning flags
CFG-MSGOUT-UBX_MON_TEMP_SPI	`0x20910335`	U1	-	-	Temperature measurement result and temperature related warning flags
CFG-MSGOUT-UBX_MON_TEMP_UART1	`0x20910332`	U1	-	-	Temperature measurement result and temperature related warning flags
CFG-MSGOUT-UBX_MON_TEMP_UART2	`0x20910333`	U1	-	-	Temperature measurement result and temperature related warning flags
CFG-MSGOUT-UBX_MON_TEMP_USB	`0x20910334`	U1	-	-	Temperature measurement result and temperature related warning flags
CFG-MSGOUT-UBX_MON_TXBUF_I2C	`0x2091019b`	U1	-	-	Output rate of the UBX-MON-TXBUF message on port I2C
CFG-MSGOUT-UBX_MON_TXBUF_SPI	`0x2091019f`	U1	-	-	Output rate of the UBX-MON-TXBUF message on port SPI
CFG-MSGOUT-UBX_MON_TXBUF_UART1	`0x2091019c`	U1	-	-	Output rate of the UBX-MON-TXBUF message on port UART1
CFG-MSGOUT-UBX_MON_TXBUF_UART2	`0x2091019d`	U1	-	-	Output rate of the UBX-MON-TXBUF message on port UART2
CFG-MSGOUT-UBX_MON_TXBUF_USB	`0x2091019e`	U1	-	-	Output rate of the UBX-MON-TXBUF message on port USB
CFG-MSGOUT-UBX_NAV_CLOCK_I2C	`0x20910065`	U1	-	-	Output rate of the UBX-NAV-CLOCK message on port I2C
CFG-MSGOUT-UBX_NAV_CLOCK_SPI	`0x20910069`	U1	-	-	Output rate of the UBX-NAV-CLOCK message on port SPI

CFG-MSGOUT- Configuration Items continued*

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Description*
CFG-MSGOUT-UBX_NAV_CLOCK_UART1	`0x20910066`	U1	-	-	Output rate of the UBX-NAV-CLOCK message on port UART1
CFG-MSGOUT-UBX_NAV_CLOCK_UART2	`0x20910067`	U1	-	-	Output rate of the UBX-NAV-CLOCK message on port UART2
CFG-MSGOUT-UBX_NAV_CLOCK_USB	`0x20910068`	U1	-	-	Output rate of the UBX-NAV-CLOCK message on port USB
CFG-MSGOUT-UBX_NAV_COV_I2C	`0x20910083`	U1	-	-	Output rate of the UBX-NAV-COV message on port I2C
CFG-MSGOUT-UBX_NAV_COV_SPI	`0x20910087`	U1	-	-	Output rate of the UBX-NAV-COV message on port SPI
CFG-MSGOUT-UBX_NAV_COV_UART1	`0x20910084`	U1	-	-	Output rate of the UBX-NAV-COV message on port UART1
CFG-MSGOUT-UBX_NAV_COV_UART2	`0x20910085`	U1	-	-	Output rate of the UBX-NAV-COV message on port UART2
CFG-MSGOUT-UBX_NAV_COV_USB	`0x20910086`	U1	-	-	Output rate of the UBX-NAV-COV message on port USB
CFG-MSGOUT-UBX_NAV_DOP_I2C	`0x20910038`	U1	-	-	Output rate of the UBX-NAV-DOP message on port I2C
CFG-MSGOUT-UBX_NAV_DOP_SPI	`0x2091003c`	U1	-	-	Output rate of the UBX-NAV-DOP message on port SPI
CFG-MSGOUT-UBX_NAV_DOP_UART1	`0x20910039`	U1	-	-	Output rate of the UBX-NAV-DOP message on port UART1
CFG-MSGOUT-UBX_NAV_DOP_UART2	`0x2091003a`	U1	-	-	Output rate of the UBX-NAV-DOP message on port UART2
CFG-MSGOUT-UBX_NAV_DOP_USB	`0x2091003b`	U1	-	-	Output rate of the UBX-NAV-DOP message on port USB
CFG-MSGOUT-UBX_NAV_EOE_I2C	`0x2091015f`	U1	-	-	Output rate of the UBX-NAV-EOE message on port I2C
CFG-MSGOUT-UBX_NAV_EOE_SPI	`0x20910163`	U1	-	-	Output rate of the UBX-NAV-EOE message on port SPI
CFG-MSGOUT-UBX_NAV_EOE_UART1	`0x20910160`	U1	-	-	Output rate of the UBX-NAV-EOE message on port UART1
CFG-MSGOUT-UBX_NAV_EOE_UART2	`0x20910161`	U1	-	-	Output rate of the UBX-NAV-EOE message on port UART2
CFG-MSGOUT-UBX_NAV_EOE_USB	`0x20910162`	U1	-	-	Output rate of the UBX-NAV-EOE message on port USB
CFG-MSGOUT-UBX_NAV_GEOFENCE_I2C	`0x209100a1`	U1	-	-	Output rate of the UBX-NAV-GEOFENCE message on port I2C
CFG-MSGOUT-UBX_NAV_GEOFENCE_SPI	`0x209100a5`	U1	-	-	Output rate of the UBX-NAV-GEOFENCE message on port SPI
CFG-MSGOUT-UBX_NAV_GEOFENCE_UART1	`0x209100a2`	U1	-	-	Output rate of the UBX-NAV-GEOFENCE message on port UART1
CFG-MSGOUT-UBX_NAV_GEOFENCE_UART2	`0x209100a3`	U1	-	-	Output rate of the UBX-NAV-GEOFENCE message on port UART2

CFG-MSGOUT- Configuration Items continued*

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Description*
CFG-MSGOUT-UBX_NAV_GEOFENCE_USB	0x209100a4	U1	-	-	Output rate of the UBX-NAV-GEOFENCE message on port USB
CFG-MSGOUT-UBX_NAV_HPPOSECEF_I2C	0x2091002e	U1	-	-	Output rate of the UBX-NAV-HPPOSECEF message on port I2C
CFG-MSGOUT-UBX_NAV_HPPOSECEF_SPI	0x20910032	U1	-	-	Output rate of the UBX-NAV-HPPOSECEF message on port SPI
CFG-MSGOUT-UBX_NAV_HPPOSECEF_UART1	0x2091002f	U1	-	-	Output rate of the UBX-NAV-HPPOSECEF message on port UART1
CFG-MSGOUT-UBX_NAV_HPPOSECEF_UART2	0x20910030	U1	-	-	Output rate of the UBX-NAV-HPPOSECEF message on port UART2
CFG-MSGOUT-UBX_NAV_HPPOSECEF_USB	0x20910031	U1	-	-	Output rate of the UBX-NAV-HPPOSECEF message on port USB
CFG-MSGOUT-UBX_NAV_HPPOSLLH_I2C	0x20910033	U1	-	-	Output rate of the UBX-NAV-HPPOSLLH message on port I2C
CFG-MSGOUT-UBX_NAV_HPPOSLLH_SPI	0x20910037	U1	-	-	Output rate of the UBX-NAV-HPPOSLLH message on port SPI
CFG-MSGOUT-UBX_NAV_HPPOSLLH_UART1	0x20910034	U1	-	-	Output rate of the UBX-NAV-HPPOSLLH message on port UART1
CFG-MSGOUT-UBX_NAV_HPPOSLLH_UART2	0x20910035	U1	-	-	Output rate of the UBX-NAV-HPPOSLLH message on port UART2
CFG-MSGOUT-UBX_NAV_HPPOSLLH_USB	0x20910036	U1	-	-	Output rate of the UBX-NAV-HPPOSLLH message on port USB
CFG-MSGOUT-UBX_NAV_ODO_I2C	0x2091007e	U1	-	-	Output rate of the UBX-NAV-ODO message on port I2C
CFG-MSGOUT-UBX_NAV_ODO_SPI	0x20910082	U1	-	-	Output rate of the UBX-NAV-ODO message on port SPI
CFG-MSGOUT-UBX_NAV_ODO_UART1	0x2091007f	U1	-	-	Output rate of the UBX-NAV-ODO message on port UART1
CFG-MSGOUT-UBX_NAV_ODO_UART2	0x20910080	U1	-	-	Output rate of the UBX-NAV-ODO message on port UART2
CFG-MSGOUT-UBX_NAV_ODO_USB	0x20910081	U1	-	-	Output rate of the UBX-NAV-ODO message on port USB
CFG-MSGOUT-UBX_NAV_ORB_I2C	0x20910010	U1	-	-	Output rate of the UBX-NAV-ORB message on port I2C
CFG-MSGOUT-UBX_NAV_ORB_SPI	0x20910014	U1	-	-	Output rate of the UBX-NAV-ORB message on port SPI
CFG-MSGOUT-UBX_NAV_ORB_UART1	0x20910011	U1	-	-	Output rate of the UBX-NAV-ORB message on port UART1
CFG-MSGOUT-UBX_NAV_ORB_UART2	0x20910012	U1	-	-	Output rate of the UBX-NAV-ORB message on port UART2
CFG-MSGOUT-UBX_NAV_ORB_USB	0x20910013	U1	-	-	Output rate of the UBX-NAV-ORB message on port USB
CFG-MSGOUT-UBX_NAV_POSECEF_I2C	0x20910024	U1	-	-	Output rate of the UBX-NAV-POSECEF message on port I2C

CFG-MSGOUT- Configuration Items continued*

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Description*
CFG-MSGOUT-UBX_NAV_POSECEF_SPI	`0x20910028`	U1	-	-	Output rate of the UBX-NAV-POSECEF message on port SPI
CFG-MSGOUT-UBX_NAV_POSECEF_UART1	`0x20910025`	U1	-	-	Output rate of the UBX-NAV-POSECEF message on port UART1
CFG-MSGOUT-UBX_NAV_POSECEF_UART2	`0x20910026`	U1	-	-	Output rate of the UBX-NAV-POSECEF message on port UART2
CFG-MSGOUT-UBX_NAV_POSECEF_USB	`0x20910027`	U1	-	-	Output rate of the UBX-NAV-POSECEF message on port USB
CFG-MSGOUT-UBX_NAV_POSLLH_I2C	`0x20910029`	U1	-	-	Output rate of the UBX-NAV-POSLLH message on port I2C
CFG-MSGOUT-UBX_NAV_POSLLH_SPI	`0x2091002d`	U1	-	-	Output rate of the UBX-NAV-POSLLH message on port SPI
CFG-MSGOUT-UBX_NAV_POSLLH_UART1	`0x2091002a`	U1	-	-	Output rate of the UBX-NAV-POSLLH message on port UART1
CFG-MSGOUT-UBX_NAV_POSLLH_UART2	`0x2091002b`	U1	-	-	Output rate of the UBX-NAV-POSLLH message on port UART2
CFG-MSGOUT-UBX_NAV_POSLLH_USB	`0x2091002c`	U1	-	-	Output rate of the UBX-NAV-POSLLH message on port USB
CFG-MSGOUT-UBX_NAV_PVT_I2C	`0x20910006`	U1	-	-	Output rate of the UBX-NAV-PVT message on port I2C
CFG-MSGOUT-UBX_NAV_PVT_SPI	`0x2091000a`	U1	-	-	Output rate of the UBX-NAV-PVT message on port SPI
CFG-MSGOUT-UBX_NAV_PVT_UART1	`0x20910007`	U1	-	-	Output rate of the UBX-NAV-PVT message on port UART1
CFG-MSGOUT-UBX_NAV_PVT_UART2	`0x20910008`	U1	-	-	Output rate of the UBX-NAV-PVT message on port UART2
CFG-MSGOUT-UBX_NAV_PVT_USB	`0x20910009`	U1	-	-	Output rate of the UBX-NAV-PVT message on port USB
CFG-MSGOUT-UBX_NAV_RELPOSNED_I2C	`0x2091008d`	U1	-	-	Output rate of the UBX-NAV-RELPOSNED message on port I2C
CFG-MSGOUT-UBX_NAV_RELPOSNED_SPI	`0x20910091`	U1	-	-	Output rate of the UBX-NAV-RELPOSNED message on port SPI
CFG-MSGOUT-UBX_NAV_RELPOSNED_UART1	`0x2091008e`	U1	-	-	Output rate of the UBX-NAV-RELPOSNED message on port UART1
CFG-MSGOUT-UBX_NAV_RELPOSNED_UART2	`0x2091008f`	U1	-	-	Output rate of the UBX-NAV-RELPOSNED message on port UART2
CFG-MSGOUT-UBX_NAV_RELPOSNED_USB	`0x20910090`	U1	-	-	Output rate of the UBX-NAV-RELPOSNED message on port USB
CFG-MSGOUT-UBX_NAV_SAT_I2C	`0x20910015`	U1	-	-	Output rate of the UBX-NAV-SAT message on port I2C
CFG-MSGOUT-UBX_NAV_SAT_SPI	`0x20910019`	U1	-	-	Output rate of the UBX-NAV-SAT message on port SPI
CFG-MSGOUT-UBX_NAV_SAT_UART1	`0x20910016`	U1	-	-	Output rate of the UBX-NAV-SAT message on port UART1

CFG-MSGOUT-* Configuration Items continued

Configuration Item	Key ID	Type	Scale	Unit	Description
CFG-MSGOUT-UBX_NAV_SAT_UART2	0x20910017	U1	-	-	Output rate of the UBX-NAV-SAT message on port UART2
CFG-MSGOUT-UBX_NAV_SAT_USB	0x20910018	U1	-	-	Output rate of the UBX-NAV-SAT message on port USB
CFG-MSGOUT-UBX_NAV_SIG_I2C	0x20910345	U1	-	-	Output rate of the UBX-NAV-SIG message on port I2C
CFG-MSGOUT-UBX_NAV_SIG_SPI	0x20910349	U1	-	-	Output rate of the UBX-NAV-SIG message on port SPI
CFG-MSGOUT-UBX_NAV_SIG_UART1	0x20910346	U1	-	-	Output rate of the UBX-NAV-SIG message on port UART1
CFG-MSGOUT-UBX_NAV_SIG_UART2	0x20910347	U1	-	-	Output rate of the UBX-NAV-SIG message on port UART2
CFG-MSGOUT-UBX_NAV_SIG_USB	0x20910348	U1	-	-	Output rate of the UBX-NAV-SIG message on port USB
CFG-MSGOUT-UBX_NAV_SOL_I2C	0x20910001	U1	-	-	Output rate of the UBX-NAV-SOL message on port I2C
CFG-MSGOUT-UBX_NAV_SOL_SPI	0x20910005	U1	-	-	Output rate of the UBX-NAV-SOL message on port SPI
CFG-MSGOUT-UBX_NAV_SOL_UART1	0x20910002	U1	-	-	Output rate of the UBX-NAV-SOL message on port UART1
CFG-MSGOUT-UBX_NAV_SOL_UART2	0x20910003	U1	-	-	Output rate of the UBX-NAV-SOL message on port UART2
CFG-MSGOUT-UBX_NAV_SOL_USB	0x20910004	U1	-	-	Output rate of the UBX-NAV-SOL message on port USB
CFG-MSGOUT-UBX_NAV_STATUS_I2C	0x2091001a	U1	-	-	Output rate of the UBX-NAV-STATUS message on port I2C
CFG-MSGOUT-UBX_NAV_STATUS_SPI	0x2091001e	U1	-	-	Output rate of the UBX-NAV-STATUS message on port SPI
CFG-MSGOUT-UBX_NAV_STATUS_UART1	0x2091001b	U1	-	-	Output rate of the UBX-NAV-STATUS message on port UART1
CFG-MSGOUT-UBX_NAV_STATUS_UART2	0x2091001c	U1	-	-	Output rate of the UBX-NAV-STATUS message on port UART2
CFG-MSGOUT-UBX_NAV_STATUS_USB	0x2091001d	U1	-	-	Output rate of the UBX-NAV-STATUS message on port USB
CFG-MSGOUT-UBX_NAV_SVINFO_I2C	0x2091000b	U1	-	-	Output rate of the UBX-NAV-SVINFO message on port I2C
CFG-MSGOUT-UBX_NAV_SVINFO_SPI	0x2091000f	U1	-	-	Output rate of the UBX-NAV-SVINFO message on port SPI
CFG-MSGOUT-UBX_NAV_SVINFO_UART1	0x2091000c	U1	-	-	Output rate of the UBX-NAV-SVINFO message on port UART1
CFG-MSGOUT-UBX_NAV_SVINFO_UART2	0x2091000d	U1	-	-	Output rate of the UBX-NAV-SVINFO message on port UART2
CFG-MSGOUT-UBX_NAV_SVINFO_USB	0x2091000e	U1	-	-	Output rate of the UBX-NAV-SVINFO message on port USB

CFG-MSGOUT- Configuration Items continued*

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Description*
CFG-MSGOUT-UBX_NAV_SVIN_I2C	`0x20910088`	U1	-	-	Output rate of the UBX-NAV-SVIN message on port I2C
CFG-MSGOUT-UBX_NAV_SVIN_SPI	`0x2091008c`	U1	-	-	Output rate of the UBX-NAV-SVIN message on port SPI
CFG-MSGOUT-UBX_NAV_SVIN_UART1	`0x20910089`	U1	-	-	Output rate of the UBX-NAV-SVIN message on port UART1
CFG-MSGOUT-UBX_NAV_SVIN_UART2	`0x2091008a`	U1	-	-	Output rate of the UBX-NAV-SVIN message on port UART2
CFG-MSGOUT-UBX_NAV_SVIN_USB	`0x2091008b`	U1	-	-	Output rate of the UBX-NAV-SVIN message on port USB
CFG-MSGOUT-UBX_NAV_TIMEBDS_I2C	`0x20910051`	U1	-	-	Output rate of the UBX-NAV-TIMEBDS message on port I2C
CFG-MSGOUT-UBX_NAV_TIMEBDS_SPI	`0x20910055`	U1	-	-	Output rate of the UBX-NAV-TIMEBDS message on port SPI
CFG-MSGOUT-UBX_NAV_TIMEBDS_UART1	`0x20910052`	U1	-	-	Output rate of the UBX-NAV-TIMEBDS message on port UART1
CFG-MSGOUT-UBX_NAV_TIMEBDS_UART2	`0x20910053`	U1	-	-	Output rate of the UBX-NAV-TIMEBDS message on port UART2
CFG-MSGOUT-UBX_NAV_TIMEBDS_USB	`0x20910054`	U1	-	-	Output rate of the UBX-NAV-TIMEBDS message on port USB
CFG-MSGOUT-UBX_NAV_TIMEGAL_I2C	`0x20910056`	U1	-	-	Output rate of the UBX-NAV-TIMEGAL message on port I2C
CFG-MSGOUT-UBX_NAV_TIMEGAL_SPI	`0x2091005a`	U1	-	-	Output rate of the UBX-NAV-TIMEGAL message on port SPI
CFG-MSGOUT-UBX_NAV_TIMEGAL_UART1	`0x20910057`	U1	-	-	Output rate of the UBX-NAV-TIMEGAL message on port UART1
CFG-MSGOUT-UBX_NAV_TIMEGAL_UART2	`0x20910058`	U1	-	-	Output rate of the UBX-NAV-TIMEGAL message on port UART2
CFG-MSGOUT-UBX_NAV_TIMEGAL_USB	`0x20910059`	U1	-	-	Output rate of the UBX-NAV-TIMEGAL message on port USB
CFG-MSGOUT-UBX_NAV_TIMEGLO_I2C	`0x2091004c`	U1	-	-	Output rate of the UBX-NAV-TIMEGLO message on port I2C
CFG-MSGOUT-UBX_NAV_TIMEGLO_SPI	`0x20910050`	U1	-	-	Output rate of the UBX-NAV-TIMEGLO message on port SPI
CFG-MSGOUT-UBX_NAV_TIMEGLO_UART1	`0x2091004d`	U1	-	-	Output rate of the UBX-NAV-TIMEGLO message on port UART1
CFG-MSGOUT-UBX_NAV_TIMEGLO_UART2	`0x2091004e`	U1	-	-	Output rate of the UBX-NAV-TIMEGLO message on port UART2
CFG-MSGOUT-UBX_NAV_TIMEGLO_USB	`0x2091004f`	U1	-	-	Output rate of the UBX-NAV-TIMEGLO message on port USB
CFG-MSGOUT-UBX_NAV_TIMEGPS_I2C	`0x20910047`	U1	-	-	Output rate of the UBX-NAV-TIMEGPS message on port I2C
CFG-MSGOUT-UBX_NAV_TIMEGPS_SPI	`0x2091004b`	U1	-	-	Output rate of the UBX-NAV-TIMEGPS message on port SPI

CFG-MSGOUT- Configuration Items continued*

Configuration Item	Key ID	Type	Scale	Unit	Description
CFG-MSGOUT-UBX_NAV_TIMEGPS_UART1	0x20910048	U1	-	-	Output rate of the UBX-NAV-TIMEGPS message on port UART1
CFG-MSGOUT-UBX_NAV_TIMEGPS_UART2	0x20910049	U1	-	-	Output rate of the UBX-NAV-TIMEGPS message on port UART2
CFG-MSGOUT-UBX_NAV_TIMEGPS_USB	0x2091004a	U1	-	-	Output rate of the UBX-NAV-TIMEGPS message on port USB
CFG-MSGOUT-UBX_NAV_TIMELS_I2C	0x20910060	U1	-	-	Output rate of the UBX-NAV-TIMELS message on port I2C
CFG-MSGOUT-UBX_NAV_TIMELS_SPI	0x20910064	U1	-	-	Output rate of the UBX-NAV-TIMELS message on port SPI
CFG-MSGOUT-UBX_NAV_TIMELS_UART1	0x20910061	U1	-	-	Output rate of the UBX-NAV-TIMELS message on port UART1
CFG-MSGOUT-UBX_NAV_TIMELS_UART2	0x20910062	U1	-	-	Output rate of the UBX-NAV-TIMELS message on port UART2
CFG-MSGOUT-UBX_NAV_TIMELS_USB	0x20910063	U1	-	-	Output rate of the UBX-NAV-TIMELS message on port USB
CFG-MSGOUT-UBX_NAV_TIMEUTC_I2C	0x2091005b	U1	-	-	Output rate of the UBX-NAV-TIMEUTC message on port I2C
CFG-MSGOUT-UBX_NAV_TIMEUTC_SPI	0x2091005f	U1	-	-	Output rate of the UBX-NAV-TIMEUTC message on port SPI
CFG-MSGOUT-UBX_NAV_TIMEUTC_UART1	0x2091005c	U1	-	-	Output rate of the UBX-NAV-TIMEUTC message on port UART1
CFG-MSGOUT-UBX_NAV_TIMEUTC_UART2	0x2091005d	U1	-	-	Output rate of the UBX-NAV-TIMEUTC message on port UART2
CFG-MSGOUT-UBX_NAV_TIMEUTC_USB	0x2091005e	U1	-	-	Output rate of the UBX-NAV-TIMEUTC message on port USB
CFG-MSGOUT-UBX_NAV_VELECEF_I2C	0x2091003d	U1	-	-	Output rate of the UBX-NAV-VELECEF message on port I2C
CFG-MSGOUT-UBX_NAV_VELECEF_SPI	0x20910041	U1	-	-	Output rate of the UBX-NAV-VELECEF message on port SPI
CFG-MSGOUT-UBX_NAV_VELECEF_UART1	0x2091003e	U1	-	-	Output rate of the UBX-NAV-VELECEF message on port UART1
CFG-MSGOUT-UBX_NAV_VELECEF_UART2	0x2091003f	U1	-	-	Output rate of the UBX-NAV-VELECEF message on port UART2
CFG-MSGOUT-UBX_NAV_VELECEF_USB	0x20910040	U1	-	-	Output rate of the UBX-NAV-VELECEF message on port USB
CFG-MSGOUT-UBX_NAV_VELNED_I2C	0x20910042	U1	-	-	Output rate of the UBX-NAV-VELNED message on port I2C
CFG-MSGOUT-UBX_NAV_VELNED_SPI	0x20910046	U1	-	-	Output rate of the UBX-NAV-VELNED message on port SPI
CFG-MSGOUT-UBX_NAV_VELNED_UART1	0x20910043	U1	-	-	Output rate of the UBX-NAV-VELNED message on port UART1
CFG-MSGOUT-UBX_NAV_VELNED_UART2	0x20910044	U1	-	-	Output rate of the UBX-NAV-VELNED message on port UART2

CFG-MSGOUT- Configuration Items continued*

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Description*
CFG-MSGOUT-UBX_NAV_VELNED_USB	`0x20910045`	U1	-	-	Output rate of the UBX-NAV-VELNED message on port USB
CFG-MSGOUT-UBX_RXM_MEASX_I2C	`0x20910204`	U1	-	-	Output rate of the UBX-RXM-MEASX message on port I2C
CFG-MSGOUT-UBX_RXM_MEASX_SPI	`0x20910208`	U1	-	-	Output rate of the UBX-RXM-MEASX message on port SPI
CFG-MSGOUT-UBX_RXM_MEASX_UART1	`0x20910205`	U1	-	-	Output rate of the UBX-RXM-MEASX message on port UART1
CFG-MSGOUT-UBX_RXM_MEASX_UART2	`0x20910206`	U1	-	-	Output rate of the UBX-RXM-MEASX message on port UART2
CFG-MSGOUT-UBX_RXM_MEASX_USB	`0x20910207`	U1	-	-	Output rate of the UBX-RXM-MEASX message on port USB
CFG-MSGOUT-UBX_RXM_RAWX_I2C	`0x209102a4`	U1	-	-	Output rate of the UBX-RXM-RAWX message on port I2C
CFG-MSGOUT-UBX_RXM_RAWX_SPI	`0x209102a8`	U1	-	-	Output rate of the UBX-RXM-RAWX message on port SPI
CFG-MSGOUT-UBX_RXM_RAWX_UART1	`0x209102a5`	U1	-	-	Output rate of the UBX-RXM-RAWX message on port UART1
CFG-MSGOUT-UBX_RXM_RAWX_UART2	`0x209102a6`	U1	-	-	Output rate of the UBX-RXM-RAWX message on port UART2
CFG-MSGOUT-UBX_RXM_RAWX_USB	`0x209102a7`	U1	-	-	Output rate of the UBX-RXM-RAWX message on port USB
CFG-MSGOUT-UBX_RXM_RLM_I2C	`0x2091025e`	U1	-	-	Output rate of the UBX-RXM-RLM message on port I2C
CFG-MSGOUT-UBX_RXM_RLM_SPI	`0x20910262`	U1	-	-	Output rate of the UBX-RXM-RLM message on port SPI
CFG-MSGOUT-UBX_RXM_RLM_UART1	`0x2091025f`	U1	-	-	Output rate of the UBX-RXM-RLM message on port UART1
CFG-MSGOUT-UBX_RXM_RLM_UART2	`0x20910260`	U1	-	-	Output rate of the UBX-RXM-RLM message on port UART2
CFG-MSGOUT-UBX_RXM_RLM_USB	`0x20910261`	U1	-	-	Output rate of the UBX-RXM-RLM message on port USB
CFG-MSGOUT-UBX_RXM_RTC5_I2C	`0x20910155`	U1	-	-	Output rate of the UBX-RXM-RTC5 message on port I2C
CFG-MSGOUT-UBX_RXM_RTC5_SPI	`0x20910159`	U1	-	-	Output rate of the UBX-RXM-RTC5 message on port SPI
CFG-MSGOUT-UBX_RXM_RTC5_UART1	`0x20910156`	U1	-	-	Output rate of the UBX-RXM-RTC5 message on port UART1
CFG-MSGOUT-UBX_RXM_RTC5_UART2	`0x20910157`	U1	-	-	Output rate of the UBX-RXM-RTC5 message on port UART2
CFG-MSGOUT-UBX_RXM_RTC5_USB	`0x20910158`	U1	-	-	Output rate of the UBX-RXM-RTC5 message on port USB
CFG-MSGOUT-UBX_RXM_RTCM_I2C	`0x20910268`	U1	-	-	Output rate of the UBX-RXM-RTCM message on port I2C

CFG-MSGOUT- Configuration Items continued*

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Description*
CFG-MSGOUT-UBX_RXM_RTCM_SPI	`0x2091026c`	U1	-	-	Output rate of the UBX-RXM-RTCM message on port SPI
CFG-MSGOUT-UBX_RXM_RTCM_UART1	`0x20910269`	U1	-	-	Output rate of the UBX-RXM-RTCM message on port UART1
CFG-MSGOUT-UBX_RXM_RTCM_UART2	`0x2091026a`	U1	-	-	Output rate of the UBX-RXM-RTCM message on port UART2
CFG-MSGOUT-UBX_RXM_RTCM_USB	`0x2091026b`	U1	-	-	Output rate of the UBX-RXM-RTCM message on port USB
CFG-MSGOUT-UBX_RXM_SFRBX_I2C	`0x20910231`	U1	-	-	Output rate of the UBX-RXM-SFRBX message on port I2C
CFG-MSGOUT-UBX_RXM_SFRBX_SPI	`0x20910235`	U1	-	-	Output rate of the UBX-RXM-SFRBX message on port SPI
CFG-MSGOUT-UBX_RXM_SFRBX_UART1	`0x20910232`	U1	-	-	Output rate of the UBX-RXM-SFRBX message on port UART1
CFG-MSGOUT-UBX_RXM_SFRBX_UART2	`0x20910233`	U1	-	-	Output rate of the UBX-RXM-SFRBX message on port UART2
CFG-MSGOUT-UBX_RXM_SFRBX_USB	`0x20910234`	U1	-	-	Output rate of the UBX-RXM-SFRBX message on port USB
CFG-MSGOUT-UBX_RXM_SVSI_I2C	`0x20910150`	U1	-	-	Output rate of the UBX-RXM-SVSI message on port I2C
CFG-MSGOUT-UBX_RXM_SVSI_SPI	`0x20910154`	U1	-	-	Output rate of the UBX-RXM-SVSI message on port SPI
CFG-MSGOUT-UBX_RXM_SVSI_UART1	`0x20910151`	U1	-	-	Output rate of the UBX-RXM-SVSI message on port UART1
CFG-MSGOUT-UBX_RXM_SVSI_UART2	`0x20910152`	U1	-	-	Output rate of the UBX-RXM-SVSI message on port UART2
CFG-MSGOUT-UBX_RXM_SVSI_USB	`0x20910153`	U1	-	-	Output rate of the UBX-RXM-SVSI message on port USB
CFG-MSGOUT-UBX_TIM_TM2_I2C	`0x20910178`	U1	-	-	Output rate of the UBX-TIM-TM2 message on port I2C
CFG-MSGOUT-UBX_TIM_TM2_SPI	`0x2091017c`	U1	-	-	Output rate of the UBX-TIM-TM2 message on port SPI
CFG-MSGOUT-UBX_TIM_TM2_UART1	`0x20910179`	U1	-	-	Output rate of the UBX-TIM-TM2 message on port UART1
CFG-MSGOUT-UBX_TIM_TM2_UART2	`0x2091017a`	U1	-	-	Output rate of the UBX-TIM-TM2 message on port UART2
CFG-MSGOUT-UBX_TIM_TM2_USB	`0x2091017b`	U1	-	-	Output rate of the UBX-TIM-TM2 message on port USB
CFG-MSGOUT-UBX_TIM_TP_I2C	`0x2091017d`	U1	-	-	Output rate of the UBX-TIM-TP message on port I2C
CFG-MSGOUT-UBX_TIM_TP_SPI	`0x20910181`	U1	-	-	Output rate of the UBX-TIM-TP message on port SPI
CFG-MSGOUT-UBX_TIM_TP_UART1	`0x2091017e`	U1	-	-	Output rate of the UBX-TIM-TP message on port UART1

CFG-MSGOUT- Configuration Items continued*

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Description*
CFG-MSGOUT-UBX_TIM_TP_UART2	`0x2091017f`	U1	-	-	Output rate of the UBX-TIM-TP message on port UART2
CFG-MSGOUT-UBX_TIM_TP_USB	`0x20910180`	U1	-	-	Output rate of the UBX-TIM-TP message on port USB
CFG-MSGOUT-UBX_TIM_VRFY_I2C	`0x20910092`	U1	-	-	Output rate of the UBX-TIM-VRFY message on port I2C
CFG-MSGOUT-UBX_TIM_VRFY_SPI	`0x20910096`	U1	-	-	Output rate of the UBX-TIM-VRFY message on port SPI
CFG-MSGOUT-UBX_TIM_VRFY_UART1	`0x20910093`	U1	-	-	Output rate of the UBX-TIM-VRFY message on port UART1
CFG-MSGOUT-UBX_TIM_VRFY_UART2	`0x20910094`	U1	-	-	Output rate of the UBX-TIM-VRFY message on port UART2
CFG-MSGOUT-UBX_TIM_VRFY_USB	`0x20910095`	U1	-	-	Output rate of the UBX-TIM-VRFY message on port USB

7.8.12 CFG-NAVHPG: High Precision Navigation Configuration

This group configures items related to the operation of the receiver in high precision, for example Differential correction and other related features.

CFG-NAVHPG-* Configuration Items

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Description*
CFG-NAVHPG-DGNSSMODE	`0x20140011`	E1	-	-	Differential corrections mode
See Constants for CFG-NAVHPG-DGNSSMODE below for a list of possible constants for this item.					

Constants for CFG-NAVHPG-DGNSSMODE

Constant	*Value*	*Description*
RTK_FLOAT	2	No attempts made to fix ambiguities
RTK_FIXED	3	Ambiguities are fixed whenever possible

7.8.13 CFG-NAVSPG: Standard Precision Navigation Configuration

This group contains configuration items related to the operation of the receiver at standard precision, including configuring postition fix mode, ionospheric model selection and other related items.

CFG-NAVSPG-* Configuration Items

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Description*
CFG-NAVSPG-FIXMODE	`0x20110011`	E1	-	-	Position fix mode
See Constants for CFG-NAVSPG-FIXMODE below for a list of possible constants for this item.					
CFG-NAVSPG-INIFIX3D	`0x10110013`	L	-	-	Initial fix must be a 3d fix
CFG-NAVSPG-WKNROLLOVER	`0x30110017`	U2	-	-	GPS week rollover number
GPS week numbers will be set correctly from this week up to 1024 weeks after this week. Range is from 1 to 4096.					
CFG-NAVSPG-USE_PPP	`0x10110019`	L	-	-	Use Precise Point Positioning
Only available with the PPP product variant.					
CFG-NAVSPG-UTCSTANDARD	`0x2011001c`	E1	-	-	UTC standard to be used

CFG-NAVSPG- Configuration Items continued*

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Description*
See also GNSS time bases. . See Constants for CFG-NAVSPG-UTCSTANDARD below for a list of possible constants for this item.					
CFG-NAVSPG-DYNMODEL	`0x20110021`	E1	-	-	Dynamic platform model
See Constants for CFG-NAVSPG-DYNMODEL below for a list of possible constants for this item.					
CFG-NAVSPG-ACKAIDING	`0x10110025`	L	-	-	Acknowledge assistance input messages
CFG-NAVSPG-USE_USRDAT	`0x10110061`	L	-	-	Use user geodetic datum parameters
This must be set together with all CFG-NAVSPG-USERDAT_* parameters.					
CFG-NAVSPG-USRDAT_MAJA	`0x50110062`	R8	-	m	Geodetic datum semi-major axis
Accepted range is from 6,300,000.0 to 6,500,000.0 meters This will only be used if CFG-NAVSPG-USE_USERDAT is set. It must be set together with all other CFG-NAVSPG-USERDAT_* parameters.					
CFG-NAVSPG-USRDAT_FLAT	`0x50110063`	R8	-	-	Geodetic datum 1.0 / flattening
Accepted range is 0.0 to 500.0. This will only be used if CFG-NAVSPG-USE_USERDAT is set. It must be set together with all other CFG-NAVSPG-USERDAT_* parameters.					
CFG-NAVSPG-USRDAT_DX	`0x40110064`	R4	-	m	Geodetic datum X axis shift at the origin
Accepted range is +/- 5000.0 meters. This will only be used if CFG-NAVSPG-USE_USERDAT is set. It must be set together with all other CFG-NAVSPG-USERDAT_* parameters.					
CFG-NAVSPG-USRDAT_DY	`0x40110065`	R4	-	m	Geodetic datum Y axis shift at the origin
Accepted range is +/- 5000.0 meters. This will only be used if CFG-NAVSPG-USE_USERDAT is set. It must be set together with all other CFG-NAVSPG-USERDAT_* parameters.					
CFG-NAVSPG-USRDAT_DZ	`0x40110066`	R4	-	m	Geodetic datum Z axis shift at the origin
Accepted range is +/- 5000.0 meters. This will only be used if CFG-NAVSPG-USE_USERDAT is set. It must be set together with all other CFG-NAVSPG-USERDAT_* parameters.					
CFG-NAVSPG-USRDAT_ROTX	`0x40110067`	R4	-	arcsec	Geodetic datum rotation about the X axis
Accepted range is +/- 20.0 milli arc seconds. This will only be used if CFG-NAVSPG-USE_USERDAT is set. It must be set together with all other CFG-NAVSPG-USERDAT_* parameters.					
CFG-NAVSPG-USRDAT_ROTY	`0x40110068`	R4	-	arcsec	Geodetic datum rotation about the Y axis ()
Accepted range is +/- 20.0 milli-arc seconds. This will only be used if CFG-NAVSPG-USE_USERDAT is set. It must be set together with all other CFG-NAVSPG-USERDAT_* parameters.					
CFG-NAVSPG-USRDAT_ROTZ	`0x40110069`	R4	-	arcsec	Geodetic datum rotation about the Z axis

CFG-NAVSPG- Configuration Items continued*

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Description*
Accepted range is +/- 20.0 milli-arc seconds. This will only be used if CFG-NAVSPG-USE_USERDAT is set. It must be set together with all other CFG-NAVSPG-USERDAT_* parameters.					
CFG-NAVSPG-USRDAT_SCALE	`0x4011006a`	R4	-	ppm	Geodetic datum scale factor
Accepted range is 0.0 to 50.0 parts per million. This will only be used if CFG-NAVSPG-USE_USERDAT is set. It must be set together with all other CFG-NAVSPG-USERDAT_* parameters.					
CFG-NAVSPG-INFIL_MINSVS	`0x201100a1`	U1	-	-	Minimum number of satellites for navigation
CFG-NAVSPG-INFIL_MAXSVS	`0x201100a2`	U1	-	-	Maximum number of satellites for navigation
CFG-NAVSPG-INFIL_MINCNO	`0x201100a3`	U1	-	dBHz	Minimum satellite signal level for navigation
CFG-NAVSPG-INFIL_MINELEV	`0x201100a4`	I1	-	deg	Minimum elevation for a GNSS satellite to be used in navigation
CFG-NAVSPG-INFIL_NCNOTHRS	`0x201100aa`	U1	-	-	Number of satellites required to have C/N0 above CFG-NAVSPG-INFIL_CNOTHRS for a fix to be attempted
.					
CFG-NAVSPG-INFIL_CNOTHRS	`0x201100ab`	U1	-	-	C/N0 threshold for deciding whether to attempt a fix
.					
CFG-NAVSPG-OUTFIL_PDOP	`0x301100b1`	U2	0.1	-	Output filter position DOP mask (threshold)
CFG-NAVSPG-OUTFIL_TDOP	`0x301100b2`	U2	0.1	-	Output filter time DOP mask (threshold)
CFG-NAVSPG-OUTFIL_PACC	`0x301100b3`	U2	-	m	Output filter position accuracy mask (threshold)
CFG-NAVSPG-OUTFIL_TACC	`0x301100b4`	U2	-	m	Output filter time accuracy mask (threshold)
CFG-NAVSPG-OUTFIL_FACC	`0x301100b5`	U2	0.01	m/s	Output filter frequency accuracy mask (threshold)
CFG-NAVSPG-CONSTR_ALT	`0x401100c1`	I4	0.01	m	Fixed altitude (mean sea level) for 2D fix mode
CFG-NAVSPG-CONSTR_ALTVAR	`0x401100c2`	U4	0.0001	m^2	Fixed altitude variance for 2D mode
CFG-NAVSPG-CONSTR_DGNSSTO	`0x201100c4`	U1	-	s	DGNSS timeout

Constants for CFG-NAVSPG-FIXMODE

Constant	*Value*	*Description*
2DONLY	1	2d only
3DONLY	2	3d only
AUTO	3	Auto 2d/3d

Constants for CFG-NAVSPG-UTCSTANDARD

Constant	*Value*	*Description*
AUTO	0	Automatic; receiver selects based on GNSS configuration
USNO	3	UTC as operated by the U.S. Naval Observatory (USNO); derived from GPS time
SU	6	UTC as operated by the former Soviet Union; derived from GLONASS time
NTSC	7	UTC as operated by the National Time Service Center, China; derived from BeiDou time

Constants for CFG-NAVSPG-DYNMODEL

Constant	*Value*	*Description*
PORT	0	Portable
STAT	2	Stationary
PED	3	Pedestrian
AUTOMOT	4	Automotive
SEA	5	Sea
AIR1	6	Airborne with <1g acceleration
AIR2	7	Airborne with <2g acceleration
AIR4	8	Airborne with <4g acceleration
WRIST	9	Wrist worn watch
.		

7.8.14 CFG-NMEA: NMEA Protocol Configuration

Configures the NMEA protocol. See section NMEA Protocol Configuration for a detailed description of the configuration effects on NMEA output.

CFG-NMEA-* Configuration Items

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Description*
CFG-NMEA-PROTVER	`0x20930001`	E1	-	-	NMEA protocol version
See Constants for CFG-NMEA-PROTVER below for a list of possible constants for this item.					
CFG-NMEA-MAXSVS	`0x20930002`	E1	-	-	Maximum number of SVs to report per Talker ID
See Constants for CFG-NMEA-MAXSVS below for a list of possible constants for this item.					
CFG-NMEA-COMPAT	`0x10930003`	L	-	-	Enable compatibility mode
This might be needed for certain applications, e.g. for an NMEA parser that expects a fixed number of digits in position coordinates.					
CFG-NMEA-CONSIDER	`0x10930004`	L	-	-	Enable considering mode
This will affect NMEA output used satellite count. If set, also considered satellites (e.g. RAIMED) are counted as used satellites as well.					
CFG-NMEA-LIMIT82	`0x10930005`	L	-	-	Enable strict limit to 82 characters maximum NMEA message length
CFG-NMEA-HIGHPREC	`0x10930006`	L	-	-	Enable high precision mode
This flag cannot be set in conjunction with either CFG-NMEA-COMPAT or CFG-NMEA-LIMIT82 Mode. .					
CFG-NMEA-SVNUMBERING	`0x20930007`	E1	-	-	Display configuration for SVs that do not have value defined in NMEA

CFG-NMEA- Configuration Items continued*

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Description*
Configures the display of satellites that do not have an NMEA-defined value. Note: this does not apply to satellites with an unknown ID. See also Satellite Numbering. See Constants for CFG-NMEA-SVNUMBERING below for a list of possible constants for this item.					
CFG-NMEA-FILT_GPS	`0x10930011`	L	-	-	Disable reporting of GPS satellites
CFG-NMEA-FILT_SBAS	`0x10930012`	L	-	-	Disable reporting of SBAS satellites
CFG-NMEA-FILT_QZSS	`0x10930015`	L	-	-	Disable reporting of QZSS satellites
CFG-NMEA-FILT_GLO	`0x10930016`	L	-	-	Disable reporting of GLONASS satellites
CFG-NMEA-FILT_BDS	`0x10930017`	L	-	-	Disable reporting of BeiDou satellites
CFG-NMEA-OUT_INVFIX	`0x10930021`	L	-	-	Enable position output for failed or invalid fixes
CFG-NMEA-OUT_MSKFIX	`0x10930022`	L	-	-	Enable position output for invalid fixes
CFG-NMEA-OUT_INVTIME	`0x10930023`	L	-	-	Enable time output for invalid times
CFG-NMEA-OUT_INVDATE	`0x10930024`	L	-	-	Enable date output for invalid dates
CFG-NMEA-OUT_ONLYGPS	`0x10930025`	L	-	-	Restrict output to GPS satellites only
CFG-NMEA-OUT_FROZENCOG	`0x10930026`	L	-	-	Enable course over ground output even if it is frozen
CFG-NMEA-MAINTALKERID	`0x20930031`	E1	-	-	Main Talker ID
By default the main Talker ID (i.e. the Talker ID used for all messages other than GSV) is determined by the GNSS assignment of the receiver's channels (see `UBX-CFG-GNSS`). This field enables the main Talker ID to be overridden. See Constants for CFG-NMEA-MAINTALKERID below for a list of possible constants for this item.					
CFG-NMEA-GSVTALKERID	`0x20930032`	E1	-	-	Talker ID for GSV NMEA messages
By default the Talker ID for GSV messages is GNSS specific (as defined by NMEA). This field enables the GSV Talker ID to be overridden. See Constants for CFG-NMEA-GSVTALKERID below for a list of possible constants for this item.					
CFG-NMEA-BDSTALKERID	`0x30930033`	U2	-	-	BeiDou Talker ID
Sets the two ASCII characters that should be used for the BeiDou Talker ID. If these are set to zero, the default BeiDou TalkerId will be used.					

Constants for CFG-NMEA-PROTVER

Constant	*Value*	*Description*
V21	`21`	NMEA protocol version 2.1
V23	`23`	NMEA protocol version 2.3
V40	`40`	NMEA protocol version 4.0
.		
V41	`41`	NMEA protocol version 4.1
.		

Constants for CFG-NMEA-MAXSVS

Constant	*Value*	*Description*
UNLIM	`0`	Unlimited
8SVS	`8`	8 SVs
12SVS	`12`	12 SVs
16SVS	`16`	16 SVs

Constants for CFG-NMEA-SVNUMBERING

Constant	*Value*	*Description*
STRICT	`0`	Strict - satellites are not output
EXTENDED	`1`	Extended - use proprietary numbering

Constants for CFG-NMEA-MAINTALKERID

Constant	*Value*	*Description*
AUTO	`0`	Main Talker ID is not overridden
GP	`1`	Set main Talker ID to 'GP'
GL	`2`	Set main Talker ID to 'GL'
GN	`3`	Set main Talker ID to 'GN'
GA	`4`	Set main Talker ID to 'GA'
.		
GB	`5`	Set main Talker ID to 'GB'
.		

Constants for CFG-NMEA-GSVTALKERID

Constant	*Value*	*Description*
GNSS	`0`	Use GNSS specific Talker ID (as defined by NMEA)
MAIN	`1`	Use the main Talker ID

7.8.15 CFG-ODO: Odometer and Low-Speed Course Over Ground Filter Configuration

The items in this group allow the user to configure the Odometer feature and Low-Speed Course Over Ground Filter.

CFG-ODO-* Configuration Items

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Description*
CFG-ODO-USE_ODO	`0x10220001`	L	-	-	Use odometer
CFG-ODO-USE_COG	`0x10220002`	L	-	-	Use low-speed course over ground filter
CFG-ODO-OUTLPVEL	`0x10220003`	L	-	-	Output low-pass filtered velocity
CFG-ODO-OUTLPCOG	`0x10220004`	L	-	-	Output low-pass filtered course over ground (heading)
CFG-ODO-PROFILE	`0x20220005`	E1	-	-	Odometer profile configuration
See Constants for CFG-ODO-PROFILE below for a list of possible constants for this item.					
CFG-ODO-COGMAXSPEED	`0x20220021`	U1	-	m/s	Upper speed limit for low-speed course over ground filter
CFG-ODO-COGMAXPOSACC	`0x20220022`	U1	-	-	Maximum acceptable position accuracy for computing low-speed filtered course over ground
CFG-ODO-VELLPGAIN	`0x20220031`	U1	-	-	Velocity low-pass filter level
Range is from 0 to 255.					
CFG-ODO-COGLPGAIN	`0x20220032`	U1	-	-	Course over ground low-pass filter level (at speed < 8 m/s)
Range is from 0 to 255.					

Constants for CFG-ODO-PROFILE

Constant	*Value*	*Description*
RUN	`0`	Running
CYCL	`1`	Cycling
SWIM	`2`	Swimming
CAR	`3`	Car
CUSTOM	`4`	Custom

7.8.16 CFG-RATE: Navigation and Measurement Rate Configuration

The configuration items in this group allow the user to alter the rate at which navigation solutions (and the measurements that they depend on) are generated by the receiver. The calculation of the navigation solution will always be aligned to the top of a second zero (first second of the week) of the configured reference time system.

For protocol version 18 and later the navigation period is an integer multiple of the measurement period.

CFG-RATE-* Configuration Items

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Description*
CFG-RATE-MEAS	`0x30210001`	U2	0.001	s	Nominal time between GNSS measurements (e.g. 100ms results in 10Hz measurement rate, 1000ms = 1Hz measurement rate)
CFG-RATE-NAV	`0x30210002`	U2	-	-	Ratio of number of measurements to number of navigation solutions
E.g. 5 means five measurements for every navigation solution. The maximum value is 127. .					
CFG-RATE-TIMEREF	`0x20210003`	E1	-	-	Time system to which measurements are aligned
See Constants for CFG-RATE-TIMEREF below for a list of possible constants for this item.					

Constants for CFG-RATE-TIMEREF

Constant	*Value*	*Description*
UTC	`0`	Align measurements to UTC time
GPS	`1`	Align measurements to GPS time
GLO	`2`	Align measurements to GLONASS time
.		
BDS	`3`	Align measurements to BeiDou time
.		
GAL	`4`	Align measurements to Galileo time
.		

7.8.17 CFG-RINV: Remote Inventory

The Remote Inventory enables storing user-defined data in the non-volatile memory of the receiver. The data can be either binary or a string of ASCII characters. In the latter case, it can optionally be output at startup after the boot screen.

CFG-RINV-* Configuration Items

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Description*
CFG-RINV-DUMP	`0x10c70001`	L	-	-	Dump data at startup
When true, data will be dumped to the interface on startup, unless CFG-RINV-BINARY is set.					
CFG-RINV-BINARY	`0x10c70002`	L	-	-	Data is binary
When true, the data is treated as binary data.					
CFG-RINV-DATA_SIZE	`0x20c70003`	U1	-	-	Size of data
Size of data to store/be stored in the Remote Inventory (maximum 30 bytes).					
CFG-RINV-CHUNK0	`0x50c70004`	X8	-	-	Data bytes 1-8 (LSB)
Data to store/be stored in Remote Inventory - max 8 bytes, left-most in LSB, e.g. string ABCD will appear as 0x44434241.					
CFG-RINV-CHUNK1	`0x50c70005`	X8	-	-	Data bytes 9-16
Data to store/be stored in Remote Inventory - max 8 bytes, left-most in LSB, e.g. string ABCD will appear as 0x44434241.					
CFG-RINV-CHUNK2	`0x50c70006`	X8	-	-	Data bytes 17-24
Data to store/be stored in Remote Inventory - max 8 bytes, left-most in LSB, e.g. string ABCD will appear as 0x44434241.					
CFG-RINV-CHUNK3	`0x50c70007`	X8	-	-	Data bytes 25-30 (MSB)
Data to store/be stored in Remote Inventory - max 6 bytes, left-most in LSB, e.g. string ABCD will appear as 0x44434241.					

7.8.18 CFG-SIGNAL: Satellite Systems (GNSS) Signal Configuration

It is necessary for at least one signal and constellation from a major GNSS to be enabled, after applying the new configuration to the current one.

The individual signals enable keys are governed by their corresponding constellation enable key. See GNSS Signal Configuration for more details.

Configuration specific to a GNSS system can be done via other groups (e.g. **CFG-SBAS-*.**)

Note that changes to any items within this group will trigger a reset to the GNSS subsystem.

CFG-SIGNAL-* Configuration Items

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Description*
CFG-SIGNAL-GPS_ENA	`0x1031001f`	L	-	-	GPS enable
CFG-SIGNAL-GPS_L1CA_ENA	`0x10310001`	L	-	-	GPS L1C/A
CFG-SIGNAL-GPS_L2C_ENA	`0x10310003`	L	-	-	GPS L2C (only on u-blox 9 F platform products)
CFG-SIGNAL-GAL_ENA	`0x10310021`	L	-	-	Galileo enable
CFG-SIGNAL-GAL_E1_ENA	`0x10310007`	L	-	-	Galileo E1
CFG-SIGNAL-GAL_E5B_ENA	`0x1031000a`	L	-	-	Galileo E5b (only on u-blox 9 F platform products)
CFG-SIGNAL-BDS_ENA	`0x10310022`	L	-	-	BeiDou Enable
CFG-SIGNAL-BDS_B1_ENA	`0x1031000d`	L	-	-	BeiDou B1I
CFG-SIGNAL-BDS_B2_ENA	`0x1031000e`	L	-	-	BeiDou B2I (only on u-blox 9 F platform products)
CFG-SIGNAL-QZSS_ENA	`0x10310024`	L	-	-	QZSS enable
CFG-SIGNAL-QZSS_L1CA_ENA	`0x10310012`	L	-	-	QZSS L1C/A
CFG-SIGNAL-QZSS_L2C_ENA	`0x10310015`	L	-	-	QZSS L2C (only on u-blox 9 F platform products)

CFG-SIGNAL-* Configuration Items continued

Configuration Item	Key ID	Type	Scale	Unit	Description
CFG-SIGNAL-GLO_ENA	0x10310025	L	-	-	GLONASS enable
CFG-SIGNAL-GLO_L1_ENA	0x10310018	L	-	-	GLONASS L1
CFG-SIGNAL-GLO_L2_ENA	0x1031001a	L	-	-	GLONASS L2 (only on u-blox 9 F platform products)

7.8.19 CFG-SPI: Configuration of the SPI Interface

Settings needed to configure the SPI communication interface.

CFG-SPI-* Configuration Items

Configuration Item	Key ID	Type	Scale	Unit	Description
CFG-SPI-MAXFF	0x20640001	U1	-	-	Number of bytes containing 0xFF to receive before switching off reception. Range: 0(mechanism off)-63
CFG-SPI-CPOLARITY	0x10640002	L	-	-	Clock polarity select: 0: Active Hight Clock, SCLK idles low, 1: Active Low Clock, SCLK idles high
CFG-SPI-CPHASE	0x10640003	L	-	-	Clock phase select: 0: Data captured on first edge of SCLK, 1: Data captured on second edge of SCLK
CFG-SPI-EXTENDEDTIMEOUT	0x10640005	L	-	-	Flag to disable timeouting the interface after 1.5s
CFG-SPI-ENABLED	0x10640006	L	-	-	Flag to indicate if the SPI interface should be enabled

7.8.20 CFG-SPIINPROT: Input Protocol Configuration of the SPI Interface

Input protocol enable flags of the SPI interface.

CFG-SPIINPROT-* Configuration Items

Configuration Item	Key ID	Type	Scale	Unit	Description
CFG-SPIINPROT-UBX	0x10790001	L	-	-	Flag to indicate if UBX should be an input protocol on SPI
CFG-SPIINPROT-NMEA	0x10790002	L	-	-	Flag to indicate if NMEA should be an input protocol on SPI
CFG-SPIINPROT-RTCM3X	0x10790004	L	-	-	Flag to indicate if RTCM3X should be an input protocol on SPI

7.8.21 CFG-SPIOUTPROT: Output Protocol Configuration of the SPI Interface

Output protocol enable flags of the SPI interface.

CFG-SPIOUTPROT-* Configuration Items

Configuration Item	Key ID	Type	Scale	Unit	Description
CFG-SPIOUTPROT-UBX	0x107a0001	L	-	-	Flag to indicate if UBX should be an output protocol on SPI
CFG-SPIOUTPROT-NMEA	0x107a0002	L	-	-	Flag to indicate if NMEA should be an output protocol on SPI

CFG-SPIOUTPROT- Configuration Items continued*

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Description*
CFG-SPIOUTPROT-RTCM3X	`0x107a0004`	L	-	-	Flag to indicate if RTCM3X should be an output protocol on SPI

7.8.22 CFG-TMODE: Time Mode Configuration

Configuration for operation of the receiver in Time Mode. The position referred to in the configuration items is that of the Antenna Reference Point (ARP).

CFG-TMODE-* Configuration Items

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Description*
CFG-TMODE-MODE	`0x20030001`	E1	-	-	Receiver mode
See Constants for CFG-TMODE-MODE below for a list of possible constants for this item.					
CFG-TMODE-POS_TYPE	`0x20030002`	E1	-	-	Determines whether the ARP position is given in ECEF or LAT/LON/HEIGHT?
See Constants for CFG-TMODE-POS_TYPE below for a list of possible constants for this item.					
CFG-TMODE-ECEF_X	`0x40030003`	I4	-	cm	ECEF X coordinate of the ARP position.
This will only be used if CFG-TMODE-MODE=FIXED and CFG-TMODE-POS_TYPE=ECEF.					
CFG-TMODE-ECEF_Y	`0x40030004`	I4	-	cm	ECEF Y coordinate of the ARP position.
This will only be used if CFG-TMODE-MODE=FIXED and CFG-TMODE-POS_TYPE=ECEF.					
CFG-TMODE-ECEF_Z	`0x40030005`	I4	-	cm	ECEF Z coordinate of the ARP position.
This will only be used if CFG-TMODE-MODE=FIXED and CFG-TMODE-POS_TYPE=ECEF.					
CFG-TMODE-ECEF_X_HP	`0x20030006`	I1	0.1	mm	High-precision ECEF X coordinate of the ARP position.
Accepted range is -99 to +99. This will only be used if CFG-TMODE-MODE=FIXED and CFG-TMODE-POS_TYPE=ECEF.					
CFG-TMODE-ECEF_Y_HP	`0x20030007`	I1	0.1	mm	High-precision ECEF Y coordinate of the ARP position.
Accepted range is -99 to +99. This will only be used if CFG-TMODE-MODE=FIXED and CFG-TMODE-POS_TYPE=ECEF.					
CFG-TMODE-ECEF_Z_HP	`0x20030008`	I1	0.1	mm	High-precision ECEF Z coordinate of the ARP position.
Accepted range is -99 to +99. This will only be used if CFG-TMODE-MODE=FIXED and CFG-TMODE-POS_TYPE=ECEF.					
CFG-TMODE-LAT	`0x40030009`	I4	1e-7	deg	Latitude of the ARP position.
This will only be used if CFG-TMODE-MODE=FIXED and CFG-TMODE-POS_TYPE=LLH.					
CFG-TMODE-LON	`0x4003000a`	I4	1e-7	deg	Longitude of the ARP position.
This will only be used if CFG-TMODE-MODE=FIXED and CFG-TMODE-POS_TYPE=LLH.					
CFG-TMODE-HEIGHT	`0x4003000b`	I4	-	cm	Height of the ARP position.
This will only be used if CFG-TMODE-MODE=FIXED and CFG-TMODE-POS_TYPE=LLH.					
CFG-TMODE-LAT_HP	`0x2003000c`	I1	1e-9	deg	High-precision latitude of the ARP position
Accepted range is -99 to +99. This will only be used if CFG-TMODE-MODE=FIXED and CFG-TMODE-POS_TYPE=LLH.					

CFG-TMODE- Configuration Items continued*

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Description*
CFG-TMODE-LON_HP	`0x2003000d`	I1	1e-9	deg	High-precision longitude of the ARP position.
Accepted range is -99 to +99. This will only be used if CFG-TMODE-MODE=FIXED and CFG-TMODE-POS_TYPE=LLH.					
CFG-TMODE-HEIGHT_HP	`0x2003000e`	I1	0.1	mm	High-precision height of the ARP position.
Accepted range is -99 to +99. This will only be used if CFG-TMODE-MODE=FIXED and CFG-TMODE-POS_TYPE=LLH.					
CFG-TMODE-FIXED_POS_ACC	`0x4003000f`	U4	0.1	mm	Fixed position 3D accuracy
CFG-TMODE-SVIN_MIN_DUR	`0x40030010`	U4	-	s	Survey-in minimum duration
This will only be used if CFG-TMODE-MODE=SURVEY_IN.					
CFG-TMODE-SVIN_ACC_LIMIT	`0x40030011`	U4	0.1	mm	Survey-in position accuracy limit
This will only be used if CFG-TMODE-MODE=SURVEY_IN.					

Constants for CFG-TMODE-MODE

Constant	*Value*	*Description*
DISABLED	`0`	Disabled
SURVEY_IN	`1`	Survey In
FIXED	`2`	Fixed Mode (true ARP position information required)

Constants for CFG-TMODE-POS_TYPE

Constant	*Value*	*Description*
ECEF	`0`	Position is ECEF
LLH	`1`	Position is Lat/Lon/Height

7.8.23 CFG-TXREADY: Tx-Ready Configuration

Configuration of the tx ready pin.

CFG-TXREADY-* Configuration Items

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Description*
CFG-TXREADY-ENABLED	`0x10a20001`	L	-	-	Flag to indicate if tx ready pin mechanism should be enabled
CFG-TXREADY-POLARITY	`0x10a20002`	L	-	-	The polarity of the tx ready pin: false: high-active, true:low-active
CFG-TXREADY-PIN	`0x20a20003`	U1	-	-	Pin number to use for the tx ready functionality
CFG-TXREADY-THRESHOLD	`0x30a20004`	U2	-	-	Amount of data that should be ready on the interface before triggering the tx ready pin
CFG-TXREADY-INTERFACE	`0x20a20005`	E1	-	-	Interface where the tx ready feature should be linked to
See Constants for CFG-TXREADY-INTERFACE below for a list of possible constants for this item.					

Constants for CFG-TXREADY-INTERFACE

Constant	*Value*	*Description*
I2C	`0`	I2C interface

Constants for CFG-TXREADY-INTERFACE continued

Constant	*Value*	*Description*
SPI	1	SPI interface

7.8.24 CFG-UART1: Configuration of the UART1 Interface

Settings needed to configure the UART1 communication interface.

CFG-UART1-* Configuration Items

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Description*
CFG-UART1-BAUDRATE	`0x40520001`	U4	-	-	The baud rate that should be configured on the UART1
CFG-UART1-STOPBITS	`0x20520002`	E1	-	-	Number of stopbits that should be used on UART1
See Constants for CFG-UART1-STOPBITS below for a list of possible constants for this item.					
CFG-UART1-DATABITS	`0x20520003`	E1	-	-	Number of databits that should be used on UART1
See Constants for CFG-UART1-DATABITS below for a list of possible constants for this item.					
CFG-UART1-PARITY	`0x20520004`	E1	-	-	Parity mode that should be used on UART1
See Constants for CFG-UART1-PARITY below for a list of possible constants for this item.					
CFG-UART1-ENABLED	`0x10520005`	L	-	-	Flag to indicate if the UART1 should be enabled
CFG-UART1-REMAP	`0x10520006`	L	-	-	UART1 Remapping

Constants for CFG-UART1-STOPBITS

Constant	*Value*	*Description*
HALF	0	0.5 stopbits
ONE	1	1.0 stopbits
ONEHALF	2	1.5 stopbits
TWO	3	2.0 stopbits

Constants for CFG-UART1-DATABITS

Constant	*Value*	*Description*
EIGHT	0	8 databits
SEVEN	1	7 databits

Constants for CFG-UART1-PARITY

Constant	*Value*	*Description*
NONE	0	No parity bit
ODD	1	Add an odd parity bit
EVEN	2	Add an even parity bit

7.8.25 CFG-UART1INPROT: Input Protocol Configuration of the UART1 Interface

Input protocol enable flags of the UART1 interface.

CFG-UART1INPROT-* Configuration Items

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Description*

CFG-UART1INPROT- Configuration Items continued*

Configuration Item	Key ID	Type	Scale	Unit	Description
CFG-UART1INPROT-UBX	0x10730001	L	-	-	Flag to indicate if UBX should be an input protocol on UART1
CFG-UART1INPROT-NMEA	0x10730002	L	-	-	Flag to indicate if NMEA should be an input protocol on UART1
CFG-UART1INPROT-RTCM3X	0x10730004	L	-	-	Flag to indicate if RTCM3X should be an input protocol on UART1

7.8.26 CFG-UART1OUTPROT: Output Protocol Configuration of the UART1 Interface

Output protocol enable flags of the UART1 interface.

CFG-UART1OUTPROT-* Configuration Items

Configuration Item	Key ID	Type	Scale	Unit	Description
CFG-UART1OUTPROT-UBX	0x10740001	L	-	-	Flag to indicate if UBX should be an output protocol on UART1
CFG-UART1OUTPROT-NMEA	0x10740002	L	-	-	Flag to indicate if NMEA should be an output protocol on UART1
CFG-UART1OUTPROT-RTCM3X	0x10740004	L	-	-	Flag to indicate if RTCM3X should be an output protocol on UART1

7.8.27 CFG-UART2: Configuration of the UART2 Interface

Settings needed to configure the UART2 communication interface.

CFG-UART2-* Configuration Items

Configuration Item	Key ID	Type	Scale	Unit	Description
CFG-UART2-BAUDRATE	0x40530001	U4	-	-	The baud rate that should be configured on the UART2
CFG-UART2-STOPBITS	0x20530002	E1	-	-	Number of stopbits that should be used on UART2
See Constants for CFG-UART2-STOPBITS below for a list of possible constants for this item.					
CFG-UART2-DATABITS	0x20530003	E1	-	-	Number of databits that should be used on UART2
See Constants for CFG-UART2-DATABITS below for a list of possible constants for this item.					
CFG-UART2-PARITY	0x20530004	E1	-	-	Parity mode that should be used on UART2
See Constants for CFG-UART2-PARITY below for a list of possible constants for this item.					
CFG-UART2-ENABLED	0x10530005	L	-	-	Flag to indicate if the UART2 should be enabled
CFG-UART2-REMAP	0x10530006	L	-	-	UART2 Remapping

Constants for CFG-UART2-STOPBITS

Constant	Value	Description
HALF	0	0.5 stopbits
ONE	1	1.0 stopbits
ONEHALF	2	1.5 stopbits
TWO	3	2.0 stopbits

Constants for CFG-UART2-DATABITS

Constant	*Value*	*Description*
EIGHT	0	8 databits
SEVEN	1	7 databits

Constants for CFG-UART2-PARITY

Constant	*Value*	*Description*
NONE	0	No parity bit
ODD	1	Add an odd parity bit
EVEN	2	Add an even parity bit

7.8.28 CFG-UART2INPROT: Input Protocol Configuration of the UART2 Interface

Input protocol enable flags of the UART2 interface.

CFG-UART2INPROT-* Configuration Items

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Description*
CFG-UART2INPROT-UBX	`0x10750001`	L	-	-	Flag to indicate if UBX should be an input protocol on UART2
CFG-UART2INPROT-NMEA	`0x10750002`	L	-	-	Flag to indicate if NMEA should be an input protocol on UART2
CFG-UART2INPROT-RTCM3X	`0x10750004`	L	-	-	Flag to indicate if RTCM3X should be an input protocol on UART2

7.8.29 CFG-UART2OUTPROT: Output Protocol Configuration of the UART2 Interface

Output protocol enable flags of the UART2 interface.

CFG-UART2OUTPROT-* Configuration Items

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Description*
CFG-UART2OUTPROT-UBX	`0x10760001`	L	-	-	Flag to indicate if UBX should be an output protocol on UART2
CFG-UART2OUTPROT-NMEA	`0x10760002`	L	-	-	Flag to indicate if NMEA should be an output protocol on UART2
CFG-UART2OUTPROT-RTCM3X	`0x10760004`	L	-	-	Flag to indicate if RTCM3X should be an output protocol on UART2

7.8.30 CFG-USB: Configuration of the USB Interface

Settings needed to configure the USB communication interface.

CFG-USB-* Configuration Items

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Description*
CFG-USB-ENABLED	`0x10650001`	L	-	-	Flag to indicate if the USB interface should be enabled
CFG-USB-SELFPOW	`0x10650002`	L	-	-	Self-Powered device
CFG-USB-VENDOR_ID	`0x3065000a`	U2	-	-	Vendor ID
CFG-USB-PRODUCT_ID	`0x3065000b`	U2	-	-	Vendor ID
CFG-USB-POWER	`0x3065000c`	U2	-	mA	Power consumption
CFG-USB-VENDOR_STR0	`0x5065000d`	X8	-	-	Vendor string characters 0-7
CFG-USB-VENDOR_STR1	`0x5065000e`	X8	-	-	Vendor string characters 8-15

CFG-USB- Configuration Items continued*

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Description*
CFG-USB-VENDOR_STR2	`0x5065000f`	X8	-	-	Vendor string characters 16-23
CFG-USB-VENDOR_STR3	`0x50650010`	X8	-	-	Vendor string characters 24-31
CFG-USB-PRODUCT_STR0	`0x50650011`	X8	-	-	Product string characters 0-7
CFG-USB-PRODUCT_STR1	`0x50650012`	X8	-	-	Product string characters 8-15
CFG-USB-PRODUCT_STR2	`0x50650013`	X8	-	-	Product string characters 16-23
CFG-USB-PRODUCT_STR3	`0x50650014`	X8	-	-	Product string characters 24-31
CFG-USB-SERIAL_NO_STR0	`0x50650015`	X8	-	-	Serial number string characters 0-7
CFG-USB-SERIAL_NO_STR1	`0x50650016`	X8	-	-	Serial number string characters 8-15
CFG-USB-SERIAL_NO_STR2	`0x50650017`	X8	-	-	Serial number string characters 16-23
CFG-USB-SERIAL_NO_STR3	`0x50650018`	X8	-	-	Serial number string characters 24-31

7.8.31 CFG-USBINPROT: Input Protocol Configuration of the USB Interface

Input protocol enable flags of the USB interface.

CFG-USBINPROT-* Configuration Items

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Description*
CFG-USBINPROT-UBX	`0x10770001`	L	-	-	Flag to indicate if UBX should be an input protocol on USB
CFG-USBINPROT-NMEA	`0x10770002`	L	-	-	Flag to indicate if NMEA should be an input protocol on USB
CFG-USBINPROT-RTCM3X	`0x10770004`	L	-	-	Flag to indicate if RTCM3X should be an input protocol on USB

7.8.32 CFG-USBOUTPROT: Output Protocol Configuration of the USB Interface

Output protocol enable flags of the USB interface.

CFG-USBOUTPROT-* Configuration Items

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Description*
CFG-USBOUTPROT-UBX	`0x10780001`	L	-	-	Flag to indicate if UBX should be an output protocol on USB
CFG-USBOUTPROT-NMEA	`0x10780002`	L	-	-	Flag to indicate if NMEA should be an output protocol on USB
CFG-USBOUTPROT-RTCM3X	`0x10780004`	L	-	-	Flag to indicate if RTCM3X should be an output protocol on USB

7.9 Legacy UBX Message Fields Reference

The following table lists the legacy UBX message fields and the corresponding Configuration Item. Note that the mapping from `UBX-CFG` message fields to Configuration Items is not necessarily 1:1.

UBX Messages Fields and the Corresponding Configuration Items

UBX Message and Field Name	*Configuration Item*
`UBX-CFG-ANT.ocd`	`CFG-HW-ANT_CFG_OPENDET`
`UBX-CFG-ANT.pdwnOnSCD`	`CFG-HW-ANT_CFG_PWRDOWN`
`UBX-CFG-ANT.pinOCD`	`CFG-HW-ANT_SUP_OPEN_PIN`
`UBX-CFG-ANT.pinSCD`	`CFG-HW-ANT_SUP_SHORT_PIN`
`UBX-CFG-ANT.pinSwitch`	`CFG-HW-ANT_SUP_SWITCH_PIN`

UBX Messages Fields and the Corresponding Configuration Items continued

UBX Message and Field Name	*Configuration Item*
UBX-CFG-ANT.recovery	CFG-HW-ANT_CFG_RECOVER
UBX-CFG-ANT.scd	CFG-HW-ANT_CFG_SHORTDET
UBX-CFG-ANT.svcs	CFG-HW-ANT_CFG_VOLTCTRL
UBX-CFG-DAT.dX	CFG-NAVSPG-USRDAT_DX
UBX-CFG-DAT.dY	CFG-NAVSPG-USRDAT_DY
UBX-CFG-DAT.dZ	CFG-NAVSPG-USRDAT_DZ
UBX-CFG-DAT.flat	CFG-NAVSPG-USRDAT_FLAT
UBX-CFG-DAT.majA	CFG-NAVSPG-USE_USRDAT
UBX-CFG-DAT.majA	CFG-NAVSPG-USRDAT_MAJA
UBX-CFG-DAT.rotX	CFG-NAVSPG-USRDAT_ROTX
UBX-CFG-DAT.rotY	CFG-NAVSPG-USRDAT_ROTY
UBX-CFG-DAT.rotZ	CFG-NAVSPG-USRDAT_ROTZ
UBX-CFG-DAT.scale	CFG-NAVSPG-USRDAT_SCALE
UBX-CFG-DGNSS.dgnssMode	CFG-NAVHPG-DGNSSMODE
UBX-CFG-GEOFENCE.confLvl	CFG-GEOFENCE-CONFLVL
UBX-CFG-GEOFENCE.lat	CFG-GEOFENCE-FENCE1_LAT
UBX-CFG-GEOFENCE.lat	CFG-GEOFENCE-FENCE2_LAT
UBX-CFG-GEOFENCE.lat	CFG-GEOFENCE-FENCE3_LAT
UBX-CFG-GEOFENCE.lat	CFG-GEOFENCE-FENCE4_LAT
UBX-CFG-GEOFENCE.lon	CFG-GEOFENCE-FENCE1_LON
UBX-CFG-GEOFENCE.lon	CFG-GEOFENCE-FENCE2_LON
UBX-CFG-GEOFENCE.lon	CFG-GEOFENCE-FENCE3_LON
UBX-CFG-GEOFENCE.lon	CFG-GEOFENCE-FENCE4_LON
UBX-CFG-GEOFENCE.numFences	CFG-GEOFENCE-USE_FENCE1
UBX-CFG-GEOFENCE.numFences	CFG-GEOFENCE-USE_FENCE2
UBX-CFG-GEOFENCE.numFences	CFG-GEOFENCE-USE_FENCE3
UBX-CFG-GEOFENCE.numFences	CFG-GEOFENCE-USE_FENCE4
UBX-CFG-GEOFENCE.pin	CFG-GEOFENCE-PIN
UBX-CFG-GEOFENCE.pinPolarity	CFG-GEOFENCE-PINPOL
UBX-CFG-GEOFENCE.pioEnabled	CFG-GEOFENCE-USE_PIO
UBX-CFG-GEOFENCE.radius	CFG-GEOFENCE-FENCE1_RAD
UBX-CFG-GEOFENCE.radius	CFG-GEOFENCE-FENCE2_RAD
UBX-CFG-GEOFENCE.radius	CFG-GEOFENCE-FENCE3_RAD
UBX-CFG-GEOFENCE.radius	CFG-GEOFENCE-FENCE4_RAD
UBX-CFG-INF.infMsgMask	CFG-INFMSG-NMEA_I2C
UBX-CFG-INF.infMsgMask	CFG-INFMSG-NMEA_SPI
UBX-CFG-INF.infMsgMask	CFG-INFMSG-NMEA_UART1
UBX-CFG-INF.infMsgMask	CFG-INFMSG-NMEA_UART2
UBX-CFG-INF.infMsgMask	CFG-INFMSG-NMEA_USB
UBX-CFG-INF.infMsgMask	CFG-INFMSG-UBX_I2C
UBX-CFG-INF.infMsgMask	CFG-INFMSG-UBX_SPI
UBX-CFG-INF.infMsgMask	CFG-INFMSG-UBX_UART1
UBX-CFG-INF.infMsgMask	CFG-INFMSG-UBX_UART2
UBX-CFG-INF.infMsgMask	CFG-INFMSG-UBX_USB
UBX-CFG-INF.protocolID	CFG-INFMSG-NMEA_I2C

UBX Messages Fields and the Corresponding Configuration Items continued

UBX Message and Field Name	*Configuration Item*
UBX-CFG-INF.protocolID	CFG-INFMSG-NMEA_SPI
UBX-CFG-INF.protocolID	CFG-INFMSG-NMEA_UART1
UBX-CFG-INF.protocolID	CFG-INFMSG-NMEA_UART2
UBX-CFG-INF.protocolID	CFG-INFMSG-NMEA_USB
UBX-CFG-INF.protocolID	CFG-INFMSG-UBX_SPI
UBX-CFG-INF.protocolID	CFG-INFMSG-UBX_UART1
UBX-CFG-INF.protocolID	CFG-INFMSG-UBX_UART2
UBX-CFG-INF.protocolID	CFG-INFMSG-UBX_USB
UBX-CFG-ITFM.antSetting	CFG-ITFM-ANTSETTING
UBX-CFG-ITFM.bbThreshold	CFG-ITFM-BBTHRESHOLD
UBX-CFG-ITFM.cwThreshold	CFG-ITFM-CWTHRESHOLD
UBX-CFG-ITFM.enable	CFG-ITFM-ENABLE
UBX-CFG-ITFM.enable2	CFG-ITFM-ENABLE_AUX
UBX-CFG-LOGFILTER.applyAllFilterSettings	CFG-LOGFILTER-APPLY_ALL_FILTERS
UBX-CFG-LOGFILTER.minInterval	CFG-LOGFILTER-MIN_INTERVAL
UBX-CFG-LOGFILTER.positionThreshold	CFG-LOGFILTER-POSITION_THRS
UBX-CFG-LOGFILTER.psmOncePerWakupEnabled	CFG-LOGFILTER-ONCE_PER_WAKE_UP_ENA
UBX-CFG-LOGFILTER.recordEnabled	CFG-LOGFILTER-RECORD_ENA
UBX-CFG-LOGFILTER.speedThreshold	CFG-LOGFILTER-SPEED_THRS
UBX-CFG-LOGFILTER.timeThreshold	CFG-LOGFILTER-TIME_THRS
UBX-CFG-NAV5.cnoThresh	CFG-NAVSPG-INFIL_CNOTHRS
UBX-CFG-NAV5.cnoThreshNumSVs	CFG-NAVSPG-INFIL_NCNOTHRS
UBX-CFG-NAV5.dgnssTimeout	CFG-NAVSPG-CONSTR_DGNSSTO
UBX-CFG-NAV5.dynModel	CFG-NAVSPG-DYNMODEL
UBX-CFG-NAV5.fixMode	CFG-NAVSPG-FIXMODE
UBX-CFG-NAV5.fixedAlt	CFG-NAVSPG-CONSTR_ALT
UBX-CFG-NAV5.fixedAltVar	CFG-NAVSPG-CONSTR_ALTVAR
UBX-CFG-NAV5.minElev	CFG-NAVSPG-INFIL_MINELEV
UBX-CFG-NAV5.pAcc	CFG-NAVSPG-OUTFIL_PACC
UBX-CFG-NAV5.pDop	CFG-NAVSPG-OUTFIL_PDOP
UBX-CFG-NAV5.staticHoldMaxDist	CFG-MOT-GNSSDIST_THRS
UBX-CFG-NAV5.staticHoldThresh	CFG-MOT-GNSSSPEED_THRS
UBX-CFG-NAV5.tAcc	CFG-NAVSPG-OUTFIL_FACC
UBX-CFG-NAV5.tAcc	CFG-NAVSPG-OUTFIL_TACC
UBX-CFG-NAV5.tDop	CFG-NAVSPG-OUTFIL_TDOP
UBX-CFG-NAV5.utcStandard	CFG-NAVSPG-UTCSTANDARD
UBX-CFG-NAVX5.ackAiding	CFG-NAVSPG-ACKAIDING
UBX-CFG-NAVX5.iniFix3D	CFG-NAVSPG-INIFIX3D
UBX-CFG-NAVX5.maxSVs	CFG-NAVSPG-INFIL_MAXSVS
UBX-CFG-NAVX5.minCNO	CFG-NAVSPG-INFIL_MINCNO
UBX-CFG-NAVX5.minSVs	CFG-NAVSPG-INFIL_MINSVS
UBX-CFG-NAVX5.usePPP	CFG-NAVSPG-USE_PPP
UBX-CFG-NAVX5.wknRollover	CFG-NAVSPG-WKNROLLOVER

UBX Messages Fields and the Corresponding Configuration Items continued

UBX Message and Field Name	*Configuration Item*
UBX-CFG-NMEA.bdsTalkerId	CFG-NMEA-BDSTALKERID
UBX-CFG-NMEA.beidou	CFG-NMEA-FILT_BDS
UBX-CFG-NMEA.compat	CFG-NMEA-COMPAT
UBX-CFG-NMEA.consider	CFG-NMEA-CONSIDER
UBX-CFG-NMEA.dateFilt	CFG-NMEA-OUT_INVDATE
UBX-CFG-NMEA.glonass	CFG-NMEA-FILT_GLO
UBX-CFG-NMEA.gps	CFG-NMEA-FILT_GPS
UBX-CFG-NMEA.gpsOnlyFilter	CFG-NMEA-OUT_ONLYGPS
UBX-CFG-NMEA.gsvTalkerId	CFG-NMEA-GSVTALKERID
UBX-CFG-NMEA.highPrec	CFG-NMEA-HIGHPREC
UBX-CFG-NMEA.limit82	CFG-NMEA-LIMIT82
UBX-CFG-NMEA.mainTalkerId	CFG-NMEA-MAINTALKERID
UBX-CFG-NMEA.mskPosFilt	CFG-NMEA-OUT_MSKFIX
UBX-CFG-NMEA.nmeaVersion	CFG-NMEA-PROTVER
UBX-CFG-NMEA.numSV	CFG-NMEA-MAXSVS
UBX-CFG-NMEA.posFilt	CFG-NMEA-OUT_INVFIX
UBX-CFG-NMEA.qzss	CFG-NMEA-FILT_QZSS
UBX-CFG-NMEA.sbas	CFG-NMEA-FILT_SBAS
UBX-CFG-NMEA.svNumbering	CFG-NMEA-SVNUMBERING
UBX-CFG-NMEA.timeFilt	CFG-NMEA-OUT_INVTIME
UBX-CFG-NMEA.trackFilt	CFG-NMEA-OUT_FROZENCOG
UBX-CFG-ODO.cogLpGain	CFG-ODO-COGLPGAIN
UBX-CFG-ODO.cogMaxPosAcc	CFG-ODO-COGMAXPOSACC
UBX-CFG-ODO.cogMaxSpeed	CFG-ODO-COGMAXSPEED
UBX-CFG-ODO.outLPCog	CFG-ODO-OUTLPCOG
UBX-CFG-ODO.outLPVel	CFG-ODO-OUTLPVEL
UBX-CFG-ODO.profile	CFG-ODO-PROFILE
UBX-CFG-ODO.useCOG	CFG-ODO-USE_COG
UBX-CFG-ODO.useODO	CFG-ODO-USE_ODO
UBX-CFG-ODO.velLpGain	CFG-ODO-VELLPGAIN
UBX-CFG-OTP.payload	CFG-HW-DCDC_DIS
UBX-CFG-OTP.payload	CFG-HW-OSC_TYPE
UBX-CFG-OTP.payload	CFG-HW-SINGLE_CLK
UBX-CFG-OTP.payload	CFG-UART1-REMAP
UBX-CFG-OTP.fileID	CFG-CLOCK-OSC_FREQ
UBX-CFG-OTP.maxCalibDeviation	CFG-HW-CLK_MAX_CALIB_DEV
UBX-CFG-OTP.maxCalibDeviationInvalid	CFG-HW-CLK_MAX_CALIB_DEV_VALID
UBX-CFG-OTP.offset	CFG-HW-CLK_OFFSET
UBX-CFG-OTP.offsetInvalid	CFG-HW-CLK_OFFSET_VALID
UBX-CFG-OTP.precision	CFG-HW-CLK_PRECISION
UBX-CFG-PRT.en	CFG-TXREADY-ENABLED
UBX-CFG-PRT.extendedTxTimeout	CFG-I2C-EXTENDEDTIMEOUT
UBX-CFG-PRT.inNmea	CFG-I2CINPROT-NMEA
UBX-CFG-PRT.inProtoMask	CFG-I2C-ENABLED
UBX-CFG-PRT.inRtcm3	CFG-I2CINPROT-RTCM3X

UBX Messages Fields and the Corresponding Configuration Items continued

UBX Message and Field Name	*Configuration Item*
UBX-CFG-PRT.inUbx	CFG-I2CINPROT-UBX
UBX-CFG-PRT.outNmea	CFG-I2COUTPROT-NMEA
UBX-CFG-PRT.outProtoMask	CFG-I2C-ENABLED
UBX-CFG-PRT.outRtcm3	CFG-I2COUTPROT-RTCM3X
UBX-CFG-PRT.outUbx	CFG-I2COUTPROT-UBX
UBX-CFG-PRT.pin	CFG-TXREADY-PIN
UBX-CFG-PRT.pol	CFG-TXREADY-POLARITY
UBX-CFG-PRT.slaveAddr	CFG-I2C-ADDRESS
UBX-CFG-PRT.thres	CFG-TXREADY-THRESHOLD
UBX-CFG-PRT.en	CFG-TXREADY-ENABLED
UBX-CFG-PRT.extendedTxTimeout	CFG-SPI-EXTENDEDTIMEOUT
UBX-CFG-PRT.ffCnt	CFG-SPI-MAXFF
UBX-CFG-PRT.inNmea	CFG-SPIINPROT-NMEA
UBX-CFG-PRT.inProtoMask	CFG-SPI-ENABLED
UBX-CFG-PRT.inRtcm3	CFG-SPIINPROT-RTCM3X
UBX-CFG-PRT.inUbx	CFG-SPIINPROT-UBX
UBX-CFG-PRT.outNmea	CFG-SPIOUTPROT-NMEA
UBX-CFG-PRT.outProtoMask	CFG-SPI-ENABLED
UBX-CFG-PRT.outRtcm3	CFG-SPIOUTPROT-RTCM3X
UBX-CFG-PRT.outUbx	CFG-SPIOUTPROT-UBX
UBX-CFG-PRT.pin	CFG-TXREADY-PIN
UBX-CFG-PRT.pol	CFG-TXREADY-POLARITY
UBX-CFG-PRT.spiMode	CFG-SPI-CPHASE
UBX-CFG-PRT.spiMode	CFG-SPI-CPOLARITY
UBX-CFG-PRT.thres	CFG-TXREADY-THRESHOLD
UBX-CFG-PRT.baudRate	CFG-UART1-BAUDRATE
UBX-CFG-PRT.baudRate	CFG-UART2-BAUDRATE
UBX-CFG-PRT.charLen	CFG-UART1-DATABITS
UBX-CFG-PRT.charLen	CFG-UART2-DATABITS
UBX-CFG-PRT.inNmea	CFG-UART1INPROT-NMEA
UBX-CFG-PRT.inNmea	CFG-UART2INPROT-NMEA
UBX-CFG-PRT.inProtoMask	CFG-UART1-ENABLED
UBX-CFG-PRT.inProtoMask	CFG-UART2-ENABLED
UBX-CFG-PRT.inRtcm3	CFG-UART1INPROT-RTCM3X
UBX-CFG-PRT.inRtcm3	CFG-UART2INPROT-RTCM3X
UBX-CFG-PRT.inUbx	CFG-UART1INPROT-UBX
UBX-CFG-PRT.inUbx	CFG-UART2INPROT-UBX
UBX-CFG-PRT.nStopBits	CFG-UART1-STOPBITS
UBX-CFG-PRT.nStopBits	CFG-UART2-STOPBITS
UBX-CFG-PRT.outNmea	CFG-UART1OUTPROT-NMEA
UBX-CFG-PRT.outNmea	CFG-UART2OUTPROT-NMEA
UBX-CFG-PRT.outProtoMask	CFG-UART1-ENABLED
UBX-CFG-PRT.outProtoMask	CFG-UART2-ENABLED
UBX-CFG-PRT.outRtcm3	CFG-UART1OUTPROT-RTCM3X
UBX-CFG-PRT.outRtcm3	CFG-UART2OUTPROT-RTCM3X

UBX Messages Fields and the Corresponding Configuration Items continued

UBX Message and Field Name	*Configuration Item*
UBX-CFG-PRT.outUbx	CFG-UART1OUTPROT-UBX
UBX-CFG-PRT.outUbx	CFG-UART2OUTPROT-UBX
UBX-CFG-PRT.parity	CFG-UART1-PARITY
UBX-CFG-PRT.parity	CFG-UART2-PARITY
UBX-CFG-PRT.inNmea	CFG-USBINPROT-NMEA
UBX-CFG-PRT.inProtoMask	CFG-USB-ENABLED
UBX-CFG-PRT.inRtcm3	CFG-USBINPROT-RTCM3X
UBX-CFG-PRT.inUbx	CFG-USBINPROT-UBX
UBX-CFG-PRT.outNmea	CFG-USBOUTPROT-NMEA
UBX-CFG-PRT.outProtoMask	CFG-USB-ENABLED
UBX-CFG-PRT.outRtcm3	CFG-USBOUTPROT-RTCM3X
UBX-CFG-PRT.outUbx	CFG-USBOUTPROT-UBX
UBX-CFG-RATE.measRate	CFG-RATE-MEAS
UBX-CFG-RATE.navRate	CFG-RATE-NAV
UBX-CFG-RATE.timeRef	CFG-RATE-TIMEREF
UBX-CFG-RINV.data	CFG-RINV-CHUNK0
UBX-CFG-RINV.data	CFG-RINV-CHUNK1
UBX-CFG-RINV.data	CFG-RINV-CHUNK2
UBX-CFG-RINV.data	CFG-RINV-CHUNK3
UBX-CFG-RINV.data	CFG-RINV-DATA_SIZE
UBX-CFG-RINV.flags	CFG-RINV-BINARY
UBX-CFG-RINV.flags	CFG-RINV-DUMP
UBX-CFG-TMODE3.ecefXOrLat	CFG-TMODE-ECEF_X
UBX-CFG-TMODE3.ecefXOrLat	CFG-TMODE-LAT
UBX-CFG-TMODE3.ecefXOrLatHP	CFG-TMODE-ECEF_X_HP
UBX-CFG-TMODE3.ecefXOrLatHP	CFG-TMODE-LAT_HP
UBX-CFG-TMODE3.ecefYOrLon	CFG-TMODE-ECEF_Y
UBX-CFG-TMODE3.ecefYOrLon	CFG-TMODE-LON
UBX-CFG-TMODE3.ecefYOrLonHP	CFG-TMODE-ECEF_Y_HP
UBX-CFG-TMODE3.ecefYOrLonHP	CFG-TMODE-LON_HP
UBX-CFG-TMODE3.ecefZOrAlt	CFG-TMODE-ECEF_Z
UBX-CFG-TMODE3.ecefZOrAlt	CFG-TMODE-HEIGHT
UBX-CFG-TMODE3.ecefZOrAltHP	CFG-TMODE-ECEF_Z_HP
UBX-CFG-TMODE3.ecefZOrAltHP	CFG-TMODE-HEIGHT_HP
UBX-CFG-TMODE3.fixedPosAcc	CFG-TMODE-FIXED_POS_ACC
UBX-CFG-TMODE3.flags	CFG-TMODE-MODE
UBX-CFG-TMODE3.flags	CFG-TMODE-POS_TYPE
UBX-CFG-TMODE3.svinAccLimit	CFG-TMODE-SVIN_ACC_LIMIT
UBX-CFG-TMODE3.svinMinDur	CFG-TMODE-SVIN_MIN_DUR
UBX-CFG-USB.powerConsumption	CFG-USB-POWER
UBX-CFG-USB.powerMode	CFG-USB-SELFPOW
UBX-CFG-USB.productID	CFG-USB-PRODUCT_ID
UBX-CFG-USB.productString	CFG-USB-PRODUCT_STR0
UBX-CFG-USB.productString	CFG-USB-PRODUCT_STR1
UBX-CFG-USB.productString	CFG-USB-PRODUCT_STR2

UBX Messages Fields and the Corresponding Configuration Items continued

UBX Message and Field Name	*Configuration Item*
UBX-CFG-USB.productString	CFG-USB-PRODUCT_STR3
UBX-CFG-USB.serialNumber	CFG-USB-SERIAL_NO_STR0
UBX-CFG-USB.serialNumber	CFG-USB-SERIAL_NO_STR1
UBX-CFG-USB.serialNumber	CFG-USB-SERIAL_NO_STR2
UBX-CFG-USB.serialNumber	CFG-USB-SERIAL_NO_STR3
UBX-CFG-USB.vendorID	CFG-USB-VENDOR_ID
UBX-CFG-USB.vendorString	CFG-USB-VENDOR_STR0
UBX-CFG-USB.vendorString	CFG-USB-VENDOR_STR1
UBX-CFG-USB.vendorString	CFG-USB-VENDOR_STR2
UBX-CFG-USB.vendorString	CFG-USB-VENDOR_STR3

8 RTCM Protocol

8.1 RTCM3

8.1.1 Supported Messages

The following RTCM 3.3 input messages are supported:

Supported RTCM 3.3 Input Messages

Message Type	*Description*
1001	L1-only GPS RTK observations
1002	Extended L1-only GPS RTK observations
1003	L1/L2 GPS RTK observations
1004	Extended L1/L2 GPS RTK observations
1005	Stationary RTK reference station ARP
1006	Stationary RTK reference station ARP with antenna height
1007	Antenna descriptor
1009	L1-only GLONASS RTK observations
1010	Extended L1-only GLONASS RTK observations
1011	L1/L2 GLONASS RTK observations
1012	Extended L1/L2 GLONASS RTK observations
1074	GPS MSM4
1075	GPS MSM5
1077	GPS MSM7
1084	GLONASS MSM4
1085	GLONASS MSM5
1087	GLONASS MSM7
1094	Galileo MSM4
1095	Galileo MSM5
1097	Galileo MSM7
1124	BeiDou MSM4
1125	BeiDou MSM5
1127	BeiDou MSM7
1230	GLONASS code-phase biases

The following RTCM 3.3 output messages are supported:

When configuring RTCM output messages using the UBX protocol message UBX-CFG-MSG, the Class/Ids shown in the table shall be used.

Supported RTCM 3.3 Output Messages

Message Type	*Cls/ID*	*Description*
1005	0xF5 0x05	Stationary RTK reference station ARP
1074	0xF5 0x4A	GPS MSM4
1077	0xF5 0x4D	GPS MSM7
1084	0xF5 0x54	GLONASS MSM4
1087	0xF5 0x57	GLONASS MSM7
1094	0xF5 0x5E	Galileo MSM4
1097	0xF5 0x61	Galileo MSM7
1124	0xF5 0x7C	BeiDou MSM4

Supported RTCM 3.3 Output Messages continued

Message Type	*Cls/ID*	*Description*
1127	0xF5 0x7F	BeiDou MSM7
1230	0xF5 0xE6	GLONASS code-phase biases

8.1.2 Configuration

The configuration of the RTK rover and reference station is explained in the Integration Manual.

The RTCM3 protocol can be disabled/enabled on communication interfaces by means of the UBX-CFG-PRT message. By default, RTCM3 is enabled.

The configuration of the RTCM3 correction stream must be done according to the following rules:

- The RTCM3 stream must contain only one reference station message (type 1005 and type 1006) in addition to the GNSS observation messages.
- All observation messages must be broadcast at the same rate.
- The reference station ID field in the GPS, GLONASS or BeiDou observation messages must be consistent with the reference station ID field in the reference station message otherwise the rover will not be able to compute its position.
- The RTCM3 stream must contain the GLONASS code-phase biases message (type 1230) otherwise the GLONASS ambiguities can only be estimated as float, even in RTK fixed mode.
- The static reference station message (type 1005 or type 1006) does not need to be broadcast at the same rate as the observation messages but the rover will not be able to compute its position until it has received a valid reference station message.
- The RTCM3 stream should only contain one type of observation messages per constellation. When using a multi-constellation configuration, all constellations should use the same type of observation messages. Mixing RTK and MSM messages will result in undefined rover behavior.
- If the receiver is configured to output RTCM messages on several ports, they must all have the same RTCM configuration otherwise the MSM multiple message bit might not be set properly.

8.1.3 Output

RTK Rover Mode will result in following modified output:

- NMEA-GGA: The quality field will be 4 for RTK fixed and 5 for RTK float (see NMEA Positon Fix Flags). The age of differential corrections and reference station ID will be set.
- NMEA-GLL, NMEA-VTG: The posMode indicator will be D for RTK float and RTK fixed (see NMEA Positon Fix Flags).
- NMEA-RMC, NMEA-GNS: The posMode indicator will be F for RTK float and R for RTK fixed (see NMEA Positon Fix Flags).
- UBX-NAV-PVT: The carrSoln flag will be set to 1 for RTK float and 2 for RTK fixed.
- UBX-NAV-RELPOSNED: The diffSoln and refPosValid flags will be set. The carrSoln flag will be set to 1 for RTK float and 2 for RTK fixed. In moving baseline rover mode, the isMoving flag will be set, and the refPosMiss and refObsMiss flags will be set for epochs during which extrapolated reference position or observations have been used.
- UBX-NAV-SAT: The diffCorr flag will be set for satellites with valid RTCM data. The rtcmCorrUsed, prCorrUsed, and crCorrUsed flags will be set for satellites for which the RTCM corrections have been applied.
- UBX-NAV-SIG: For signals to which the RTCM corrections have been applied, the correction source will be set to RTCM3 OSR and the crUsed, prCorrUsed, and crCorrUsed flags will be set.

- UBX-NAV-STATUS: The diffSoln flag will be set; the diffCorr flag will be set.
- If the baseline exceeds 50km and a message type 1005 or type 1006 is received, a UBX-INF-WARNING will be output, e.g. "WARNING: DGNSS baseline big: 52.7km"

8.1.4 Reference

The RTCM3 support is implemented according to RTCM STANDARD 10403.3 DIFFERENTIAL GNSS (GLOBAL NAVIGATION SATELLITE SYSTEMS) SERVICES - VERSION 3.

Appendix

A Satellite Numbering

A summary of all the SV numbering schemes is provided in the following table.

Satellite numbering

GNSS Type	*SV range*	*UBX gnssId:svId*	*UBX svId*	*NMEA 2.X-4.0 (strict)*	*NMEA 2.X-4.0 (extended)*	*NMEA 4.1+ (strict)*	*NMEA 4.1+ (extended)*
GPS	G1-G32	0:1-32	1-32	1-32	1-32	1-32	1-32
SBAS	S120-S158	1:120-158	120-158	33-64	33-64,152-158	33-64	33-64,152-158
Galileo	E1-E36	2:1-36	211-246	-	301-336	1-36	1-36
BeiDou	B1-B37	3:1-37	159-163,33-64	-	401-437	1-37	1-37
QZSS	Q1-Q5	5:1-5	193-197	-	193-197	-	193-197
GLONASS	R1-R32, R?	6:1-32, 6:255	65-96, 255	65-96, null	65-96, null	65-96, null	65-96, null

B UBX and NMEA Signal Identifiers

UBX and NMEA protocols uses signal identifiers (commonly abbreviated to "sigId") to distinguish between different signals from GNSS.

Signal identifiers are only valid when combined with a GNSS identifier (see above). The table below shows the range of identifiers currently supported in the firmware.

The following table shows the mapping of GNSS signals to UBX / NMEA signal identifier.

Signal Identifiers

Signal name	*UBX gnssId*	*UBX sigId*	*NMEA 4.1+ gnssId*	*NMEA 4.1+ sigId*
GPS L1C/A*	0	0	1	1
GPS L2 CL	0	3	1	6
GPS L2 CM	0	4	1	5
Galileo E1 C*	2	0	3	7
Galileo E1 B*	2	1	3	7
Galileo E5 bI	2	5	3	2
Galileo E5 bQ	2	6	3	2
BeiDou B1I D1*	3	0	4**	1**
BeiDou B1I D2*	3	1	4**	1**
BeiDou B2I D1	3	2	4**	3**
BeiDou B2I D2	3	3	4**	3**
QZSS L1C/A*	5	0		
GLONASS L1 OF*	6	0	2	1
GLONASS L2 OF	6	2	2	3

UBX messages, that don't have an explicit sigId field, contain information about the subset of signals marked with (*).

BeiDou gnssId and sigId are not defined in the NMEA protocol version 4.1, values shown in the table are valid for u-blox products only (**).

C Configuration Defaults

The configuration defaults given in this section apply to the receiver firmwares given below.

These values assume that the defaults have not been changed using eFuse OTP or Pin configuration (see Default Layer Composite).

C.1 u-blox 9 ZED-F9P (version 1.00 HPG 1.00B03)

This section lists the configuration defaults for the *u-blox 9 ZED-F9P (version 1.00 HPG 1.00B03)*, protocol version 27.00.

System Clock Configuration (CFG-CLOCK-*) Configuration Defaults

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Default Value*
CFG-CLOCK-OSC_FREQ	0x40a4000d	U4	-	Hz	26000000

Geofencing Configuration (CFG-GEOFENCE-*) Configuration Defaults

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Default Value*
CFG-GEOFENCE-CONFLVL	0x20240011	E1	-	-	0 (L000)
CFG-GEOFENCE-USE_PIO	0x10240012	L	-	-	0 (false)
CFG-GEOFENCE-PINPOL	0x20240013	E1	-	-	0 (LOW_IN)
CFG-GEOFENCE-PIN	0x20240014	U1	-	-	0
CFG-GEOFENCE-USE_FENCE1	0x10240020	L	-	-	0 (false)
CFG-GEOFENCE-FENCE1_LAT	0x40240021	I4	1e-7	deg	0
CFG-GEOFENCE-FENCE1_LON	0x40240022	I4	1e-7	deg	0
CFG-GEOFENCE-FENCE1_RAD	0x40240023	U4	0.01	m	0
CFG-GEOFENCE-USE_FENCE2	0x10240030	L	-	-	0 (false)
CFG-GEOFENCE-FENCE2_LAT	0x40240031	I4	1e-7	deg	0
CFG-GEOFENCE-FENCE2_LON	0x40240032	I4	1e-7	deg	0
CFG-GEOFENCE-FENCE2_RAD	0x40240033	U4	0.01	m	0
CFG-GEOFENCE-USE_FENCE3	0x10240040	L	-	-	0 (false)
CFG-GEOFENCE-FENCE3_LAT	0x40240041	I4	1e-7	deg	0
CFG-GEOFENCE-FENCE3_LON	0x40240042	I4	1e-7	deg	0
CFG-GEOFENCE-FENCE3_RAD	0x40240043	U4	0.01	m	0
CFG-GEOFENCE-USE_FENCE4	0x10240050	L	-	-	0 (false)
CFG-GEOFENCE-FENCE4_LAT	0x40240051	I4	1e-7	deg	0
CFG-GEOFENCE-FENCE4_LON	0x40240052	I4	1e-7	deg	0
CFG-GEOFENCE-FENCE4_RAD	0x40240053	U4	0.01	m	0

Hardware Configuration (CFG-HW-*) Configuration Defaults

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Default Value*
CFG-HW-DCDC_DIS	0x10a30018	L	-	-	0 (false)
CFG-HW-SINGLE_CLK	0x10a30019	L	-	-	0 (false)
CFG-HW-OSC_TYPE	0x20a30025	E1	-	-	0 (TCXO_D1V2)
CFG-HW-CLK_OFFSET	0x40a30028	I4	-	ppb	0
CFG-HW-CLK_OFFSET_VALID	0x10a30029	L	-	-	0 (false)
CFG-HW-CLK_PRECISION	0x40a3002a	U4	-	ppb	0
CFG-HW-CLK_MAX_CALIB_DEV	0x40a3002b	U4	-	ppb	0
CFG-HW-CLK_MAX_CALIB_DEV_VALID	0x10a3002c	L	-	-	0 (false)
CFG-HW-ANT_CFG_VOLTCTRL	0x10a3002e	L	-	-	0 (false)

Hardware Configuration (CFG-HW-) Configuration Defaults continued*

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Default Value*
CFG-HW-ANT_CFG_SHORTDET	0x10a3002f	L	-	-	0 (false)
CFG-HW-ANT_CFG_OPENDET	0x10a30031	L	-	-	0 (false)
CFG-HW-ANT_CFG_PWRDOWN	0x10a30033	L	-	-	0 (false)
CFG-HW-ANT_CFG_RECOVER	0x10a30035	L	-	-	0 (false)
CFG-HW-ANT_SUP_SWITCH_PIN	0x20a30036	U1	-	-	16
CFG-HW-ANT_SUP_SHORT_PIN	0x20a30037	U1	-	-	15
CFG-HW-ANT_SUP_OPEN_PIN	0x20a30038	U1	-	-	8
CFG-HW-RFDC_TIMEOUT	0x20a30050	U1	-	s	0

Configuration of the I2C Interface (CFG-I2C-*) Configuration Defaults

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Default Value*
CFG-I2C-ADDRESS	0x20510001	U1	-	-	132
CFG-I2C-EXTENDEDTIMEOUT	0x10510002	L	-	-	0 (false)
CFG-I2C-ENABLED	0x10510003	L	-	-	1 (true)

Input Protocol Configuration of the I2C Interface (CFG-I2CINPROT-*) Configuration Defaults

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Default Value*
CFG-I2CINPROT-UBX	0x10710001	L	-	-	1 (true)
CFG-I2CINPROT-NMEA	0x10710002	L	-	-	1 (true)
CFG-I2CINPROT-RTCM3X	0x10710004	L	-	-	1 (true)

Output Protocol Configuration of the I2C Interface (CFG-I2COUTPROT-*) Configuration Defaults

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Default Value*
CFG-I2COUTPROT-UBX	0x10720001	L	-	-	1 (true)
CFG-I2COUTPROT-NMEA	0x10720002	L	-	-	1 (true)
CFG-I2COUTPROT-RTCM3X	0x10720004	L	-	-	1 (true)

Inf Message Configuration (CFG-INFMSG-*) Configuration Defaults

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Default Value*
CFG-INFMSG-UBX_I2C	0x20920001	X1	-	-	0x00
CFG-INFMSG-UBX_UART1	0x20920002	X1	-	-	0x00
CFG-INFMSG-UBX_UART2	0x20920003	X1	-	-	0x00
CFG-INFMSG-UBX_USB	0x20920004	X1	-	-	0x00
CFG-INFMSG-UBX_SPI	0x20920005	X1	-	-	0x00
CFG-INFMSG-NMEA_I2C	0x20920006	X1	-	-	0x07 (ERROR | WARNING | NOTICE)
CFG-INFMSG-NMEA_UART1	0x20920007	X1	-	-	0x07 (ERROR | WARNING | NOTICE)
CFG-INFMSG-NMEA_UART2	0x20920008	X1	-	-	0x07 (ERROR | WARNING | NOTICE)
CFG-INFMSG-NMEA_USB	0x20920009	X1	-	-	0x07 (ERROR | WARNING | NOTICE)
CFG-INFMSG-NMEA_SPI	0x2092000a	X1	-	-	0x07 (ERROR | WARNING | NOTICE)

Jamming/Interference Monitor configuration (CFG-ITFM-*) Configuration Defaults

Configuration Item	Key ID	Type	Scale	Unit	Default Value
CFG-ITFM-BBTHRESHOLD	0x20410001	U1	-	-	3
CFG-ITFM-CWTHRESHOLD	0x20410002	U1	-	-	15
CFG-ITFM-ENABLE	0x1041000d	L	-	-	0 (false)
CFG-ITFM-ANTSETTING	0x20410010	E1	-	-	0 (UNKNOWN)
CFG-ITFM-ENABLE_AUX	0x10410013	L	-	-	0 (false)

Data Logger Configuration (CFG-LOGFILTER-*) Configuration Defaults

Configuration Item	Key ID	Type	Scale	Unit	Default Value
CFG-LOGFILTER-RECORD_ENA	0x10de0002	L	-	-	0 (false)
CFG-LOGFILTER-ONCE_PER_WAKE_UP_ENA	0x10de0003	L	-	-	0 (false)
CFG-LOGFILTER-APPLY_ALL_FILTERS	0x10de0004	L	-	-	0 (false)
CFG-LOGFILTER-MIN_INTERVAL	0x30de0005	U2	-	s	0
CFG-LOGFILTER-TIME_THRS	0x30de0006	U2	-	s	0
CFG-LOGFILTER-SPEED_THRS	0x30de0007	U2	-	m/s	0
CFG-LOGFILTER-POSITION_THRS	0x40de0008	U4	-	m	0

Motion Detector Configuration (CFG-MOT-*) Configuration Defaults

Configuration Item	Key ID	Type	Scale	Unit	Default Value
CFG-MOT-GNSSSPEED_THRS	0x20250038	U1	0.01	m/s	0
CFG-MOT-GNSSDIST_THRS	0x3025003b	U2	-	-	0

Message Output Configuration (CFG-MSGOUT-*) Configuration Defaults

Configuration Item	Key ID	Type	Scale	Unit	Default Value
CFG-MSGOUT-NMEA_ID_DTM_I2C	0x209100a6	U1	-	-	0
CFG-MSGOUT-NMEA_ID_DTM_SPI	0x209100aa	U1	-	-	0
CFG-MSGOUT-NMEA_ID_DTM_UART1	0x209100a7	U1	-	-	0
CFG-MSGOUT-NMEA_ID_DTM_UART2	0x209100a8	U1	-	-	0
CFG-MSGOUT-NMEA_ID_DTM_USB	0x209100a9	U1	-	-	0
CFG-MSGOUT-NMEA_ID_GBS_I2C	0x209100dd	U1	-	-	0
CFG-MSGOUT-NMEA_ID_GBS_SPI	0x209100e1	U1	-	-	0
CFG-MSGOUT-NMEA_ID_GBS_UART1	0x209100de	U1	-	-	0
CFG-MSGOUT-NMEA_ID_GBS_UART2	0x209100df	U1	-	-	0
CFG-MSGOUT-NMEA_ID_GBS_USB	0x209100e0	U1	-	-	0
CFG-MSGOUT-NMEA_ID_GGA_I2C	0x209100ba	U1	-	-	1
CFG-MSGOUT-NMEA_ID_GGA_SPI	0x209100be	U1	-	-	1
CFG-MSGOUT-NMEA_ID_GGA_UART1	0x209100bb	U1	-	-	1
CFG-MSGOUT-NMEA_ID_GGA_UART2	0x209100bc	U1	-	-	1
CFG-MSGOUT-NMEA_ID_GGA_USB	0x209100bd	U1	-	-	1
CFG-MSGOUT-NMEA_ID_GLL_I2C	0x209100c9	U1	-	-	1
CFG-MSGOUT-NMEA_ID_GLL_SPI	0x209100cd	U1	-	-	1
CFG-MSGOUT-NMEA_ID_GLL_UART1	0x209100ca	U1	-	-	1
CFG-MSGOUT-NMEA_ID_GLL_UART2	0x209100cb	U1	-	-	1
CFG-MSGOUT-NMEA_ID_GLL_USB	0x209100cc	U1	-	-	1
CFG-MSGOUT-NMEA_ID_GNS_I2C	0x209100b5	U1	-	-	0
CFG-MSGOUT-NMEA_ID_GNS_SPI	0x209100b9	U1	-	-	0

Message Output Configuration (CFG-MSGOUT-) Configuration Defaults continued*

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Default Value*
CFG-MSGOUT-NMEA_ID_GNS_UART1	0x209100b6	U1	-	-	0
CFG-MSGOUT-NMEA_ID_GNS_UART2	0x209100b7	U1	-	-	0
CFG-MSGOUT-NMEA_ID_GNS_USB	0x209100b8	U1	-	-	0
CFG-MSGOUT-NMEA_ID_GRS_I2C	0x209100ce	U1	-	-	0
CFG-MSGOUT-NMEA_ID_GRS_SPI	0x209100d2	U1	-	-	0
CFG-MSGOUT-NMEA_ID_GRS_UART1	0x209100cf	U1	-	-	0
CFG-MSGOUT-NMEA_ID_GRS_UART2	0x209100d0	U1	-	-	0
CFG-MSGOUT-NMEA_ID_GRS_USB	0x209100d1	U1	-	-	0
CFG-MSGOUT-NMEA_ID_GSA_I2C	0x209100bf	U1	-	-	1
CFG-MSGOUT-NMEA_ID_GSA_SPI	0x209100c3	U1	-	-	1
CFG-MSGOUT-NMEA_ID_GSA_UART1	0x209100c0	U1	-	-	1
CFG-MSGOUT-NMEA_ID_GSA_UART2	0x209100c1	U1	-	-	1
CFG-MSGOUT-NMEA_ID_GSA_USB	0x209100c2	U1	-	-	1
CFG-MSGOUT-NMEA_ID_GST_I2C	0x209100d3	U1	-	-	0
CFG-MSGOUT-NMEA_ID_GST_SPI	0x209100d7	U1	-	-	0
CFG-MSGOUT-NMEA_ID_GST_UART1	0x209100d4	U1	-	-	0
CFG-MSGOUT-NMEA_ID_GST_UART2	0x209100d5	U1	-	-	0
CFG-MSGOUT-NMEA_ID_GST_USB	0x209100d6	U1	-	-	0
CFG-MSGOUT-NMEA_ID_GSV_I2C	0x209100c4	U1	-	-	1
CFG-MSGOUT-NMEA_ID_GSV_SPI	0x209100c8	U1	-	-	1
CFG-MSGOUT-NMEA_ID_GSV_UART1	0x209100c5	U1	-	-	1
CFG-MSGOUT-NMEA_ID_GSV_UART2	0x209100c6	U1	-	-	1
CFG-MSGOUT-NMEA_ID_GSV_USB	0x209100c7	U1	-	-	1
CFG-MSGOUT-NMEA_ID_RMC_I2C	0x209100ab	U1	-	-	1
CFG-MSGOUT-NMEA_ID_RMC_SPI	0x209100af	U1	-	-	1
CFG-MSGOUT-NMEA_ID_RMC_UART1	0x209100ac	U1	-	-	1
CFG-MSGOUT-NMEA_ID_RMC_UART2	0x209100ad	U1	-	-	1
CFG-MSGOUT-NMEA_ID_RMC_USB	0x209100ae	U1	-	-	1
CFG-MSGOUT-NMEA_ID_VLW_I2C	0x209100e7	U1	-	-	0
CFG-MSGOUT-NMEA_ID_VLW_SPI	0x209100eb	U1	-	-	0
CFG-MSGOUT-NMEA_ID_VLW_UART1	0x209100e8	U1	-	-	0
CFG-MSGOUT-NMEA_ID_VLW_UART2	0x209100e9	U1	-	-	0
CFG-MSGOUT-NMEA_ID_VLW_USB	0x209100ea	U1	-	-	0
CFG-MSGOUT-NMEA_ID_VTG_I2C	0x209100b0	U1	-	-	1
CFG-MSGOUT-NMEA_ID_VTG_SPI	0x209100b4	U1	-	-	1
CFG-MSGOUT-NMEA_ID_VTG_UART1	0x209100b1	U1	-	-	1
CFG-MSGOUT-NMEA_ID_VTG_UART2	0x209100b2	U1	-	-	1
CFG-MSGOUT-NMEA_ID_VTG_USB	0x209100b3	U1	-	-	1
CFG-MSGOUT-NMEA_ID_ZDA_I2C	0x209100d8	U1	-	-	0
CFG-MSGOUT-NMEA_ID_ZDA_SPI	0x209100dc	U1	-	-	0
CFG-MSGOUT-NMEA_ID_ZDA_UART1	0x209100d9	U1	-	-	0
CFG-MSGOUT-NMEA_ID_ZDA_UART2	0x209100da	U1	-	-	0
CFG-MSGOUT-NMEA_ID_ZDA_USB	0x209100db	U1	-	-	0
CFG-MSGOUT-PUBX_ID_POLYP_I2C	0x209100ec	U1	-	-	0
CFG-MSGOUT-PUBX_ID_POLYP_SPI	0x209100f0	U1	-	-	0

Message Output Configuration (CFG-MSGOUT-) Configuration Defaults continued*

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Default Value*
CFG-MSGOUT-PUBX_ID_POLYP_UART1	0x209100ed	U1	-	-	0
CFG-MSGOUT-PUBX_ID_POLYP_UART2	0x209100ee	U1	-	-	0
CFG-MSGOUT-PUBX_ID_POLYP_USB	0x209100ef	U1	-	-	0
CFG-MSGOUT-PUBX_ID_POLYS_I2C	0x209100f1	U1	-	-	0
CFG-MSGOUT-PUBX_ID_POLYS_SPI	0x209100f5	U1	-	-	0
CFG-MSGOUT-PUBX_ID_POLYS_UART1	0x209100f2	U1	-	-	0
CFG-MSGOUT-PUBX_ID_POLYS_UART2	0x209100f3	U1	-	-	0
CFG-MSGOUT-PUBX_ID_POLYS_USB	0x209100f4	U1	-	-	0
CFG-MSGOUT-PUBX_ID_POLYT_I2C	0x209100f6	U1	-	-	0
CFG-MSGOUT-PUBX_ID_POLYT_SPI	0x209100fa	U1	-	-	0
CFG-MSGOUT-PUBX_ID_POLYT_UART1	0x209100f7	U1	-	-	0
CFG-MSGOUT-PUBX_ID_POLYT_UART2	0x209100f8	U1	-	-	0
CFG-MSGOUT-PUBX_ID_POLYT_USB	0x209100f9	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1005_I2C	0x209102bd	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1005_SPI	0x209102c1	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1005_UART1	0x209102be	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1005_UART2	0x209102bf	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1005_USB	0x209102c0	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1077_I2C	0x209102cc	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1077_SPI	0x209102d0	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1077_UART1	0x209102cd	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1077_UART2	0x209102ce	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1077_USB	0x209102cf	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1087_I2C	0x209102d1	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1087_SPI	0x209102d5	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1087_UART1	0x209102d2	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1087_UART2	0x209102d3	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1087_USB	0x209102d4	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1097_I2C	0x20910318	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1097_SPI	0x2091031c	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1097_UART1	0x20910319	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1097_UART2	0x2091031a	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1097_USB	0x2091031b	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1127_I2C	0x209102d6	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1127_SPI	0x209102da	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1127_UART1	0x209102d7	U1	-	-	0

Message Output Configuration (CFG-MSGOUT-) Configuration Defaults continued*

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Default Value*
CFG-MSGOUT-RTCM_3X_TYPE1127_UART2	0x209102d8	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1127_USB	0x209102d9	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1230_I2C	0x20910303	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1230_SPI	0x20910307	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1230_UART1	0x20910304	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1230_UART2	0x20910305	U1	-	-	0
CFG-MSGOUT-RTCM_3X_TYPE1230_USB	0x20910306	U1	-	-	0
CFG-MSGOUT-UBX_AID_ALM_I2C	0x2091016e	U1	-	-	0
CFG-MSGOUT-UBX_AID_ALM_SPI	0x20910172	U1	-	-	0
CFG-MSGOUT-UBX_AID_ALM_UART1	0x2091016f	U1	-	-	0
CFG-MSGOUT-UBX_AID_ALM_UART2	0x20910170	U1	-	-	0
CFG-MSGOUT-UBX_AID_ALM_USB	0x20910171	U1	-	-	0
CFG-MSGOUT-UBX_AID_EPH_I2C	0x20910164	U1	-	-	0
CFG-MSGOUT-UBX_AID_EPH_SPI	0x20910168	U1	-	-	0
CFG-MSGOUT-UBX_AID_EPH_UART1	0x20910165	U1	-	-	0
CFG-MSGOUT-UBX_AID_EPH_UART2	0x20910166	U1	-	-	0
CFG-MSGOUT-UBX_AID_EPH_USB	0x20910167	U1	-	-	0
CFG-MSGOUT-UBX_AID_INI_I2C	0x209100fb	U1	-	-	0
CFG-MSGOUT-UBX_AID_INI_SPI	0x209100ff	U1	-	-	0
CFG-MSGOUT-UBX_AID_INI_UART1	0x209100fc	U1	-	-	0
CFG-MSGOUT-UBX_AID_INI_UART2	0x209100fd	U1	-	-	0
CFG-MSGOUT-UBX_AID_INI_USB	0x209100fe	U1	-	-	0
CFG-MSGOUT-UBX_LOG_INFO_I2C	0x20910259	U1	-	-	0
CFG-MSGOUT-UBX_LOG_INFO_SPI	0x2091025d	U1	-	-	0
CFG-MSGOUT-UBX_LOG_INFO_UART1	0x2091025a	U1	-	-	0
CFG-MSGOUT-UBX_LOG_INFO_UART2	0x2091025b	U1	-	-	0
CFG-MSGOUT-UBX_LOG_INFO_USB	0x2091025c	U1	-	-	0
CFG-MSGOUT-UBX_MON_HW2_I2C	0x209101b9	U1	-	-	0
CFG-MSGOUT-UBX_MON_HW2_SPI	0x209101bd	U1	-	-	0
CFG-MSGOUT-UBX_MON_HW2_UART1	0x209101ba	U1	-	-	0
CFG-MSGOUT-UBX_MON_HW2_UART2	0x209101bb	U1	-	-	0
CFG-MSGOUT-UBX_MON_HW2_USB	0x209101bc	U1	-	-	0
CFG-MSGOUT-UBX_MON_HW_I2C	0x209101b4	U1	-	-	0
CFG-MSGOUT-UBX_MON_HW_SPI	0x209101b8	U1	-	-	0
CFG-MSGOUT-UBX_MON_HW_UART1	0x209101b5	U1	-	-	0
CFG-MSGOUT-UBX_MON_HW_UART2	0x209101b6	U1	-	-	0
CFG-MSGOUT-UBX_MON_HW_USB	0x209101b7	U1	-	-	0
CFG-MSGOUT-UBX_MON_IO_I2C	0x209101a5	U1	-	-	0
CFG-MSGOUT-UBX_MON_IO_SPI	0x209101a9	U1	-	-	0
CFG-MSGOUT-UBX_MON_IO_UART1	0x209101a6	U1	-	-	0
CFG-MSGOUT-UBX_MON_IO_UART2	0x209101a7	U1	-	-	0
CFG-MSGOUT-UBX_MON_IO_USB	0x209101a8	U1	-	-	0

Message Output Configuration (CFG-MSGOUT-) Configuration Defaults continued*

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Default Value*
CFG-MSGOUT-UBX_MON_MSGPP_I2C	0x20910196	U1	-	-	0
CFG-MSGOUT-UBX_MON_MSGPP_SPI	0x2091019a	U1	-	-	0
CFG-MSGOUT-UBX_MON_MSGPP_UART1	0x20910197	U1	-	-	0
CFG-MSGOUT-UBX_MON_MSGPP_UART2	0x20910198	U1	-	-	0
CFG-MSGOUT-UBX_MON_MSGPP_USB	0x20910199	U1	-	-	0
CFG-MSGOUT-UBX_MON_PT2_I2C	0x20910209	U1	-	-	0
CFG-MSGOUT-UBX_MON_PT2_SPI	0x2091020d	U1	-	-	0
CFG-MSGOUT-UBX_MON_PT2_UART1	0x2091020a	U1	-	-	0
CFG-MSGOUT-UBX_MON_PT2_UART2	0x2091020b	U1	-	-	0
CFG-MSGOUT-UBX_MON_PT2_USB	0x2091020c	U1	-	-	0
CFG-MSGOUT-UBX_MON_RXBUF_I2C	0x209101a0	U1	-	-	0
CFG-MSGOUT-UBX_MON_RXBUF_SPI	0x209101a4	U1	-	-	0
CFG-MSGOUT-UBX_MON_RXBUF_UART1	0x209101a1	U1	-	-	0
CFG-MSGOUT-UBX_MON_RXBUF_UART2	0x209101a2	U1	-	-	0
CFG-MSGOUT-UBX_MON_RXBUF_USB	0x209101a3	U1	-	-	0
CFG-MSGOUT-UBX_MON_RXR_I2C	0x20910187	U1	-	-	0
CFG-MSGOUT-UBX_MON_RXR_SPI	0x2091018b	U1	-	-	0
CFG-MSGOUT-UBX_MON_RXR_UART1	0x20910188	U1	-	-	0
CFG-MSGOUT-UBX_MON_RXR_UART2	0x20910189	U1	-	-	0
CFG-MSGOUT-UBX_MON_RXR_USB	0x2091018a	U1	-	-	0
CFG-MSGOUT-UBX_MON_TEMP_I2C	0x20910331	U1	-	-	0
CFG-MSGOUT-UBX_MON_TEMP_SPI	0x20910335	U1	-	-	0
CFG-MSGOUT-UBX_MON_TEMP_UART1	0x20910332	U1	-	-	0
CFG-MSGOUT-UBX_MON_TEMP_UART2	0x20910333	U1	-	-	0
CFG-MSGOUT-UBX_MON_TEMP_USB	0x20910334	U1	-	-	0
CFG-MSGOUT-UBX_MON_TXBUF_I2C	0x2091019b	U1	-	-	0
CFG-MSGOUT-UBX_MON_TXBUF_SPI	0x2091019f	U1	-	-	0
CFG-MSGOUT-UBX_MON_TXBUF_UART1	0x2091019c	U1	-	-	0
CFG-MSGOUT-UBX_MON_TXBUF_UART2	0x2091019d	U1	-	-	0
CFG-MSGOUT-UBX_MON_TXBUF_USB	0x2091019e	U1	-	-	0
CFG-MSGOUT-UBX_NAV_CLOCK_I2C	0x20910065	U1	-	-	0
CFG-MSGOUT-UBX_NAV_CLOCK_SPI	0x20910069	U1	-	-	0
CFG-MSGOUT-UBX_NAV_CLOCK_UART1	0x20910066	U1	-	-	0
CFG-MSGOUT-UBX_NAV_CLOCK_UART2	0x20910067	U1	-	-	0
CFG-MSGOUT-UBX_NAV_CLOCK_USB	0x20910068	U1	-	-	0
CFG-MSGOUT-UBX_NAV_COV_I2C	0x20910083	U1	-	-	0
CFG-MSGOUT-UBX_NAV_COV_SPI	0x20910087	U1	-	-	0
CFG-MSGOUT-UBX_NAV_COV_UART1	0x20910084	U1	-	-	0
CFG-MSGOUT-UBX_NAV_COV_UART2	0x20910085	U1	-	-	0
CFG-MSGOUT-UBX_NAV_COV_USB	0x20910086	U1	-	-	0
CFG-MSGOUT-UBX_NAV_DOP_I2C	0x20910038	U1	-	-	0
CFG-MSGOUT-UBX_NAV_DOP_SPI	0x2091003c	U1	-	-	0
CFG-MSGOUT-UBX_NAV_DOP_UART1	0x20910039	U1	-	-	0
CFG-MSGOUT-UBX_NAV_DOP_UART2	0x2091003a	U1	-	-	0
CFG-MSGOUT-UBX_NAV_DOP_USB	0x2091003b	U1	-	-	0

Message Output Configuration (CFG-MSGOUT-) Configuration Defaults continued*

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Default Value*
CFG-MSGOUT-UBX_NAV_EOE_I2C	0x2091015f	U1	-	-	0
CFG-MSGOUT-UBX_NAV_EOE_SPI	0x20910163	U1	-	-	0
CFG-MSGOUT-UBX_NAV_EOE_UART1	0x20910160	U1	-	-	0
CFG-MSGOUT-UBX_NAV_EOE_UART2	0x20910161	U1	-	-	0
CFG-MSGOUT-UBX_NAV_EOE_USB	0x20910162	U1	-	-	0
CFG-MSGOUT-UBX_NAV_GEOFENCE_I2C	0x209100a1	U1	-	-	0
CFG-MSGOUT-UBX_NAV_GEOFENCE_SPI	0x209100a5	U1	-	-	0
CFG-MSGOUT-UBX_NAV_GEOFENCE_UART1	0x209100a2	U1	-	-	0
CFG-MSGOUT-UBX_NAV_GEOFENCE_UART2	0x209100a3	U1	-	-	0
CFG-MSGOUT-UBX_NAV_GEOFENCE_USB	0x209100a4	U1	-	-	0
CFG-MSGOUT-UBX_NAV_HPPOSECEF_I2C	0x2091002e	U1	-	-	0
CFG-MSGOUT-UBX_NAV_HPPOSECEF_SPI	0x20910032	U1	-	-	0
CFG-MSGOUT-UBX_NAV_HPPOSECEF_UART1	0x2091002f	U1	-	-	0
CFG-MSGOUT-UBX_NAV_HPPOSECEF_UART2	0x20910030	U1	-	-	0
CFG-MSGOUT-UBX_NAV_HPPOSECEF_USB	0x20910031	U1	-	-	0
CFG-MSGOUT-UBX_NAV_HPPOSLLH_I2C	0x20910033	U1	-	-	0
CFG-MSGOUT-UBX_NAV_HPPOSLLH_SPI	0x20910037	U1	-	-	0
CFG-MSGOUT-UBX_NAV_HPPOSLLH_UART1	0x20910034	U1	-	-	0
CFG-MSGOUT-UBX_NAV_HPPOSLLH_UART2	0x20910035	U1	-	-	0
CFG-MSGOUT-UBX_NAV_HPPOSLLH_USB	0x20910036	U1	-	-	0
CFG-MSGOUT-UBX_NAV_ODO_I2C	0x2091007e	U1	-	-	0
CFG-MSGOUT-UBX_NAV_ODO_SPI	0x20910082	U1	-	-	0
CFG-MSGOUT-UBX_NAV_ODO_UART1	0x2091007f	U1	-	-	0
CFG-MSGOUT-UBX_NAV_ODO_UART2	0x20910080	U1	-	-	0
CFG-MSGOUT-UBX_NAV_ODO_USB	0x20910081	U1	-	-	0
CFG-MSGOUT-UBX_NAV_ORB_I2C	0x20910010	U1	-	-	0
CFG-MSGOUT-UBX_NAV_ORB_SPI	0x20910014	U1	-	-	0
CFG-MSGOUT-UBX_NAV_ORB_UART1	0x20910011	U1	-	-	0
CFG-MSGOUT-UBX_NAV_ORB_UART2	0x20910012	U1	-	-	0
CFG-MSGOUT-UBX_NAV_ORB_USB	0x20910013	U1	-	-	0
CFG-MSGOUT-UBX_NAV_POSECEF_I2C	0x20910024	U1	-	-	0
CFG-MSGOUT-UBX_NAV_POSECEF_SPI	0x20910028	U1	-	-	0
CFG-MSGOUT-UBX_NAV_POSECEF_UART1	0x20910025	U1	-	-	0
CFG-MSGOUT-UBX_NAV_POSECEF_UART2	0x20910026	U1	-	-	0
CFG-MSGOUT-UBX_NAV_POSECEF_USB	0x20910027	U1	-	-	0
CFG-MSGOUT-UBX_NAV_POSLLH_I2C	0x20910029	U1	-	-	0
CFG-MSGOUT-UBX_NAV_POSLLH_SPI	0x2091002d	U1	-	-	0
CFG-MSGOUT-UBX_NAV_POSLLH_UART1	0x2091002a	U1	-	-	0
CFG-MSGOUT-UBX_NAV_POSLLH_UART2	0x2091002b	U1	-	-	0

Message Output Configuration (CFG-MSGOUT-) Configuration Defaults continued*

Configuration Item	Key ID	Type	Scale	Unit	Default Value
CFG-MSGOUT-UBX_NAV_POSLLH_USB	0x2091002c	U1	-	-	0
CFG-MSGOUT-UBX_NAV_PVT_I2C	0x20910006	U1	-	-	0
CFG-MSGOUT-UBX_NAV_PVT_SPI	0x2091000a	U1	-	-	0
CFG-MSGOUT-UBX_NAV_PVT_UART1	0x20910007	U1	-	-	0
CFG-MSGOUT-UBX_NAV_PVT_UART2	0x20910008	U1	-	-	0
CFG-MSGOUT-UBX_NAV_PVT_USB	0x20910009	U1	-	-	0
CFG-MSGOUT-UBX_NAV_RELPOSNED_I2C	0x2091008d	U1	-	-	0
CFG-MSGOUT-UBX_NAV_RELPOSNED_SPI	0x20910091	U1	-	-	0
CFG-MSGOUT-UBX_NAV_RELPOSNED_UART1	0x2091008e	U1	-	-	0
CFG-MSGOUT-UBX_NAV_RELPOSNED_UART2	0x2091008f	U1	-	-	0
CFG-MSGOUT-UBX_NAV_RELPOSNED_USB	0x20910090	U1	-	-	0
CFG-MSGOUT-UBX_NAV_SAT_I2C	0x20910015	U1	-	-	0
CFG-MSGOUT-UBX_NAV_SAT_SPI	0x20910019	U1	-	-	0
CFG-MSGOUT-UBX_NAV_SAT_UART1	0x20910016	U1	-	-	0
CFG-MSGOUT-UBX_NAV_SAT_UART2	0x20910017	U1	-	-	0
CFG-MSGOUT-UBX_NAV_SAT_USB	0x20910018	U1	-	-	0
CFG-MSGOUT-UBX_NAV_SIG_I2C	0x20910345	U1	-	-	0
CFG-MSGOUT-UBX_NAV_SIG_SPI	0x20910349	U1	-	-	0
CFG-MSGOUT-UBX_NAV_SIG_UART1	0x20910346	U1	-	-	0
CFG-MSGOUT-UBX_NAV_SIG_UART2	0x20910347	U1	-	-	0
CFG-MSGOUT-UBX_NAV_SIG_USB	0x20910348	U1	-	-	0
CFG-MSGOUT-UBX_NAV_SOL_I2C	0x20910001	U1	-	-	0
CFG-MSGOUT-UBX_NAV_SOL_SPI	0x20910005	U1	-	-	0
CFG-MSGOUT-UBX_NAV_SOL_UART1	0x20910002	U1	-	-	0
CFG-MSGOUT-UBX_NAV_SOL_UART2	0x20910003	U1	-	-	0
CFG-MSGOUT-UBX_NAV_SOL_USB	0x20910004	U1	-	-	0
CFG-MSGOUT-UBX_NAV_STATUS_I2C	0x2091001a	U1	-	-	0
CFG-MSGOUT-UBX_NAV_STATUS_SPI	0x2091001e	U1	-	-	0
CFG-MSGOUT-UBX_NAV_STATUS_UART1	0x2091001b	U1	-	-	0
CFG-MSGOUT-UBX_NAV_STATUS_UART2	0x2091001c	U1	-	-	0
CFG-MSGOUT-UBX_NAV_STATUS_USB	0x2091001d	U1	-	-	0
CFG-MSGOUT-UBX_NAV_SVINFO_I2C	0x2091000b	U1	-	-	0
CFG-MSGOUT-UBX_NAV_SVINFO_SPI	0x2091000f	U1	-	-	0
CFG-MSGOUT-UBX_NAV_SVINFO_UART1	0x2091000c	U1	-	-	0
CFG-MSGOUT-UBX_NAV_SVINFO_UART2	0x2091000d	U1	-	-	0
CFG-MSGOUT-UBX_NAV_SVINFO_USB	0x2091000e	U1	-	-	0
CFG-MSGOUT-UBX_NAV_SVIN_I2C	0x20910088	U1	-	-	0
CFG-MSGOUT-UBX_NAV_SVIN_SPI	0x2091008c	U1	-	-	0
CFG-MSGOUT-UBX_NAV_SVIN_UART1	0x20910089	U1	-	-	0
CFG-MSGOUT-UBX_NAV_SVIN_UART2	0x2091008a	U1	-	-	0
CFG-MSGOUT-UBX_NAV_SVIN_USB	0x2091008b	U1	-	-	0
CFG-MSGOUT-UBX_NAV_TIMEBDS_I2C	0x20910051	U1	-	-	0
CFG-MSGOUT-UBX_NAV_TIMEBDS_SPI	0x20910055	U1	-	-	0

Message Output Configuration (CFG-MSGOUT-) Configuration Defaults continued*

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Default Value*
CFG-MSGOUT-UBX_NAV_TIMEBDS_UART1	0x20910052	U1	-	-	0
CFG-MSGOUT-UBX_NAV_TIMEBDS_UART2	0x20910053	U1	-	-	0
CFG-MSGOUT-UBX_NAV_TIMEBDS_USB	0x20910054	U1	-	-	0
CFG-MSGOUT-UBX_NAV_TIMEGAL_I2C	0x20910056	U1	-	-	0
CFG-MSGOUT-UBX_NAV_TIMEGAL_SPI	0x2091005a	U1	-	-	0
CFG-MSGOUT-UBX_NAV_TIMEGAL_UART1	0x20910057	U1	-	-	0
CFG-MSGOUT-UBX_NAV_TIMEGAL_UART2	0x20910058	U1	-	-	0
CFG-MSGOUT-UBX_NAV_TIMEGAL_USB	0x20910059	U1	-	-	0
CFG-MSGOUT-UBX_NAV_TIMEGLO_I2C	0x2091004c	U1	-	-	0
CFG-MSGOUT-UBX_NAV_TIMEGLO_SPI	0x20910050	U1	-	-	0
CFG-MSGOUT-UBX_NAV_TIMEGLO_UART1	0x2091004d	U1	-	-	0
CFG-MSGOUT-UBX_NAV_TIMEGLO_UART2	0x2091004e	U1	-	-	0
CFG-MSGOUT-UBX_NAV_TIMEGLO_USB	0x2091004f	U1	-	-	0
CFG-MSGOUT-UBX_NAV_TIMEGPS_I2C	0x20910047	U1	-	-	0
CFG-MSGOUT-UBX_NAV_TIMEGPS_SPI	0x2091004b	U1	-	-	0
CFG-MSGOUT-UBX_NAV_TIMEGPS_UART1	0x20910048	U1	-	-	0
CFG-MSGOUT-UBX_NAV_TIMEGPS_UART2	0x20910049	U1	-	-	0
CFG-MSGOUT-UBX_NAV_TIMEGPS_USB	0x2091004a	U1	-	-	0
CFG-MSGOUT-UBX_NAV_TIMELS_I2C	0x20910060	U1	-	-	0
CFG-MSGOUT-UBX_NAV_TIMELS_SPI	0x20910064	U1	-	-	0
CFG-MSGOUT-UBX_NAV_TIMELS_UART1	0x20910061	U1	-	-	0
CFG-MSGOUT-UBX_NAV_TIMELS_UART2	0x20910062	U1	-	-	0
CFG-MSGOUT-UBX_NAV_TIMELS_USB	0x20910063	U1	-	-	0
CFG-MSGOUT-UBX_NAV_TIMEUTC_I2C	0x2091005b	U1	-	-	0
CFG-MSGOUT-UBX_NAV_TIMEUTC_SPI	0x2091005f	U1	-	-	0
CFG-MSGOUT-UBX_NAV_TIMEUTC_UART1	0x2091005c	U1	-	-	0
CFG-MSGOUT-UBX_NAV_TIMEUTC_UART2	0x2091005d	U1	-	-	0
CFG-MSGOUT-UBX_NAV_TIMEUTC_USB	0x2091005e	U1	-	-	0
CFG-MSGOUT-UBX_NAV_VELECEF_I2C	0x2091003d	U1	-	-	0
CFG-MSGOUT-UBX_NAV_VELECEF_SPI	0x20910041	U1	-	-	0
CFG-MSGOUT-UBX_NAV_VELECEF_UART1	0x2091003e	U1	-	-	0
CFG-MSGOUT-UBX_NAV_VELECEF_UART2	0x2091003f	U1	-	-	0
CFG-MSGOUT-UBX_NAV_VELECEF_USB	0x20910040	U1	-	-	0
CFG-MSGOUT-UBX_NAV_VELNED_I2C	0x20910042	U1	-	-	0
CFG-MSGOUT-UBX_NAV_VELNED_SPI	0x20910046	U1	-	-	0
CFG-MSGOUT-UBX_NAV_VELNED_UART1	0x20910043	U1	-	-	0
CFG-MSGOUT-UBX_NAV_VELNED_UART2	0x20910044	U1	-	-	0
CFG-MSGOUT-UBX_NAV_VELNED_USB	0x20910045	U1	-	-	0
CFG-MSGOUT-UBX_RXM_MEASX_I2C	0x20910204	U1	-	-	0
CFG-MSGOUT-UBX_RXM_MEASX_SPI	0x20910208	U1	-	-	0
CFG-MSGOUT-UBX_RXM_MEASX_UART1	0x20910205	U1	-	-	0
CFG-MSGOUT-UBX_RXM_MEASX_UART2	0x20910206	U1	-	-	0
CFG-MSGOUT-UBX_RXM_MEASX_USB	0x20910207	U1	-	-	0
CFG-MSGOUT-UBX_RXM_RAWX_I2C	0x209102a4	U1	-	-	0
CFG-MSGOUT-UBX_RXM_RAWX_SPI	0x209102a8	U1	-	-	0

Message Output Configuration (CFG-MSGOUT-) Configuration Defaults continued*

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Default Value*
CFG-MSGOUT-UBX_RXM_RAWX_UART1	0x209102a5	U1	-	-	0
CFG-MSGOUT-UBX_RXM_RAWX_UART2	0x209102a6	U1	-	-	0
CFG-MSGOUT-UBX_RXM_RAWX_USB	0x209102a7	U1	-	-	0
CFG-MSGOUT-UBX_RXM_RLM_I2C	0x2091025e	U1	-	-	0
CFG-MSGOUT-UBX_RXM_RLM_SPI	0x20910262	U1	-	-	0
CFG-MSGOUT-UBX_RXM_RLM_UART1	0x2091025f	U1	-	-	0
CFG-MSGOUT-UBX_RXM_RLM_UART2	0x20910260	U1	-	-	0
CFG-MSGOUT-UBX_RXM_RLM_USB	0x20910261	U1	-	-	0
CFG-MSGOUT-UBX_RXM_RTC5_I2C	0x20910155	U1	-	-	0
CFG-MSGOUT-UBX_RXM_RTC5_SPI	0x20910159	U1	-	-	0
CFG-MSGOUT-UBX_RXM_RTC5_UART1	0x20910156	U1	-	-	0
CFG-MSGOUT-UBX_RXM_RTC5_UART2	0x20910157	U1	-	-	0
CFG-MSGOUT-UBX_RXM_RTC5_USB	0x20910158	U1	-	-	0
CFG-MSGOUT-UBX_RXM_RTCM_I2C	0x20910268	U1	-	-	0
CFG-MSGOUT-UBX_RXM_RTCM_SPI	0x2091026c	U1	-	-	0
CFG-MSGOUT-UBX_RXM_RTCM_UART1	0x20910269	U1	-	-	0
CFG-MSGOUT-UBX_RXM_RTCM_UART2	0x2091026a	U1	-	-	0
CFG-MSGOUT-UBX_RXM_RTCM_USB	0x2091026b	U1	-	-	0
CFG-MSGOUT-UBX_RXM_SFRBX_I2C	0x20910231	U1	-	-	0
CFG-MSGOUT-UBX_RXM_SFRBX_SPI	0x20910235	U1	-	-	0
CFG-MSGOUT-UBX_RXM_SFRBX_UART1	0x20910232	U1	-	-	0
CFG-MSGOUT-UBX_RXM_SFRBX_UART2	0x20910233	U1	-	-	0
CFG-MSGOUT-UBX_RXM_SFRBX_USB	0x20910234	U1	-	-	0
CFG-MSGOUT-UBX_RXM_SVSI_I2C	0x20910150	U1	-	-	0
CFG-MSGOUT-UBX_RXM_SVSI_SPI	0x20910154	U1	-	-	0
CFG-MSGOUT-UBX_RXM_SVSI_UART1	0x20910151	U1	-	-	0
CFG-MSGOUT-UBX_RXM_SVSI_UART2	0x20910152	U1	-	-	0
CFG-MSGOUT-UBX_RXM_SVSI_USB	0x20910153	U1	-	-	0
CFG-MSGOUT-UBX_TIM_TM2_I2C	0x20910178	U1	-	-	0
CFG-MSGOUT-UBX_TIM_TM2_SPI	0x2091017c	U1	-	-	0
CFG-MSGOUT-UBX_TIM_TM2_UART1	0x20910179	U1	-	-	0
CFG-MSGOUT-UBX_TIM_TM2_UART2	0x2091017a	U1	-	-	0
CFG-MSGOUT-UBX_TIM_TM2_USB	0x2091017b	U1	-	-	0
CFG-MSGOUT-UBX_TIM_TP_I2C	0x2091017d	U1	-	-	0
CFG-MSGOUT-UBX_TIM_TP_SPI	0x20910181	U1	-	-	0
CFG-MSGOUT-UBX_TIM_TP_UART1	0x2091017e	U1	-	-	0
CFG-MSGOUT-UBX_TIM_TP_UART2	0x2091017f	U1	-	-	0
CFG-MSGOUT-UBX_TIM_TP_USB	0x20910180	U1	-	-	0
CFG-MSGOUT-UBX_TIM_VRFY_I2C	0x20910092	U1	-	-	0
CFG-MSGOUT-UBX_TIM_VRFY_SPI	0x20910096	U1	-	-	0
CFG-MSGOUT-UBX_TIM_VRFY_UART1	0x20910093	U1	-	-	0
CFG-MSGOUT-UBX_TIM_VRFY_UART2	0x20910094	U1	-	-	0
CFG-MSGOUT-UBX_TIM_VRFY_USB	0x20910095	U1	-	-	0

High Precision Navigation Configuration (CFG-NAVHPG-*) Configuration Defaults

Configuration Item	Key ID	Type	Scale	Unit	Default Value
CFG-NAVHPG-DGNSSMODE	0x20140011	E1	-	-	3 (RTK_FIXED)

Standard Precision Navigation Configuration (CFG-NAVSPG-*) Configuration Defaults

Configuration Item	Key ID	Type	Scale	Unit	Default Value
CFG-NAVSPG-FIXMODE	0x20110011	E1	-	-	3 (AUTO)
CFG-NAVSPG-INIFIX3D	0x10110013	L	-	-	0 (false)
CFG-NAVSPG-WKNROLLOVER	0x30110017	U2	-	-	1867
CFG-NAVSPG-USE_PPP	0x10110019	L	-	-	0 (false)
CFG-NAVSPG-UTCSTANDARD	0x2011001c	E1	-	-	0 (AUTO)
CFG-NAVSPG-DYNMODEL	0x20110021	E1	-	-	0 (PORT)
CFG-NAVSPG-ACKAIDING	0x10110025	L	-	-	0 (false)
CFG-NAVSPG-USE_USRDAT	0x10110061	L	-	-	0 (false)
CFG-NAVSPG-USRDAT_MAJA	0x50110062	R8	-	m	6378137
CFG-NAVSPG-USRDAT_FLAT	0x50110063	R8	-	-	298.257223563000002502
CFG-NAVSPG-USRDAT_DX	0x40110064	R4	-	m	0
CFG-NAVSPG-USRDAT_DY	0x40110065	R4	-	m	0
CFG-NAVSPG-USRDAT_DZ	0x40110066	R4	-	m	0
CFG-NAVSPG-USRDAT_ROTX	0x40110067	R4	-	arcsec	0
CFG-NAVSPG-USRDAT_ROTY	0x40110068	R4	-	arcsec	0
CFG-NAVSPG-USRDAT_ROTZ	0x40110069	R4	-	arcsec	0
CFG-NAVSPG-USRDAT_SCALE	0x4011006a	R4	-	ppm	0
CFG-NAVSPG-INFIL_MINSVS	0x201100a1	U1	-	-	3
CFG-NAVSPG-INFIL_MAXSVS	0x201100a2	U1	-	-	32
CFG-NAVSPG-INFIL_MINCNO	0x201100a3	U1	-	dBHz	6
CFG-NAVSPG-INFIL_MINELEV	0x201100a4	I1	-	deg	10
CFG-NAVSPG-INFIL_NCNOTHRS	0x201100aa	U1	-	-	0
CFG-NAVSPG-INFIL_CNOTHRS	0x201100ab	U1	-	-	0
CFG-NAVSPG-OUTFIL_PDOP	0x301100b1	U2	0.1	-	250
CFG-NAVSPG-OUTFIL_TDOP	0x301100b2	U2	0.1	-	250
CFG-NAVSPG-OUTFIL_PACC	0x301100b3	U2	-	m	100
CFG-NAVSPG-OUTFIL_TACC	0x301100b4	U2	-	m	350
CFG-NAVSPG-OUTFIL_FACC	0x301100b5	U2	0.01	m/s	150
CFG-NAVSPG-CONSTR_ALT	0x401100c1	I4	0.01	m	0
CFG-NAVSPG-CONSTR_ALTVAR	0x401100c2	U4	0.0001	m^2	10000
CFG-NAVSPG-CONSTR_DGNSSTO	0x201100c4	U1	-	s	60

NMEA Protocol Configuration (CFG-NMEA-*) Configuration Defaults

Configuration Item	Key ID	Type	Scale	Unit	Default Value
CFG-NMEA-PROTVER	0x20930001	E1	-	-	41 (V41)
CFG-NMEA-MAXSVS	0x20930002	E1	-	-	0 (UNLIM)
CFG-NMEA-COMPAT	0x10930003	L	-	-	0 (false)
CFG-NMEA-CONSIDER	0x10930004	L	-	-	1 (true)
CFG-NMEA-LIMIT82	0x10930005	L	-	-	0 (false)
CFG-NMEA-HIGHPREC	0x10930006	L	-	-	0 (false)

NMEA Protocol Configuration (CFG-NMEA-) Configuration Defaults continued*

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Default Value*
CFG-NMEA-SVNUMBERING	0x20930007	E1	-	-	0 (STRICT)
CFG-NMEA-FILT_GPS	0x10930011	L	-	-	0 (false)
CFG-NMEA-FILT_SBAS	0x10930012	L	-	-	0 (false)
CFG-NMEA-FILT_QZSS	0x10930015	L	-	-	0 (false)
CFG-NMEA-FILT_GLO	0x10930016	L	-	-	0 (false)
CFG-NMEA-FILT_BDS	0x10930017	L	-	-	0 (false)
CFG-NMEA-OUT_INVFIX	0x10930021	L	-	-	0 (false)
CFG-NMEA-OUT_MSKFIX	0x10930022	L	-	-	0 (false)
CFG-NMEA-OUT_INVTIME	0x10930023	L	-	-	0 (false)
CFG-NMEA-OUT_INVDATE	0x10930024	L	-	-	0 (false)
CFG-NMEA-OUT_ONLYGPS	0x10930025	L	-	-	0 (false)
CFG-NMEA-OUT_FROZENCOG	0x10930026	L	-	-	0 (false)
CFG-NMEA-MAINTALKERID	0x20930031	E1	-	-	0 (AUTO)
CFG-NMEA-GSVTALKERID	0x20930032	E1	-	-	0 (GNSS)
CFG-NMEA-BDSTALKERID	0x30930033	U2	-	-	0

Odometer and Low-Speed Course Over Ground Filter Configuration (CFG-ODO-*) Configuration Defaults

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Default Value*
CFG-ODO-USE_ODO	0x10220001	L	-	-	0 (false)
CFG-ODO-USE_COG	0x10220002	L	-	-	0 (false)
CFG-ODO-OUTLPVEL	0x10220003	L	-	-	0 (false)
CFG-ODO-OUTLPCOG	0x10220004	L	-	-	0 (false)
CFG-ODO-PROFILE	0x20220005	E1	-	-	0 (RUN)
CFG-ODO-COGMAXSPEED	0x20220021	U1	-	m/s	10
CFG-ODO-COGMAXPOSACC	0x20220022	U1	-	-	50
CFG-ODO-VELLPGAIN	0x20220031	U1	-	-	153
CFG-ODO-COGLPGAIN	0x20220032	U1	-	-	76

Navigation and Measurement Rate Configuration (CFG-RATE-*) Configuration Defaults

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Default Value*
CFG-RATE-MEAS	0x30210001	U2	0.001	s	1000
CFG-RATE-NAV	0x30210002	U2	-	-	1
CFG-RATE-TIMEREF	0x20210003	E1	-	-	1 (GPS)

Remote Inventory (CFG-RINV-*) Configuration Defaults

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Default Value*
CFG-RINV-DUMP	0x10c70001	L	-	-	0 (false)
CFG-RINV-BINARY	0x10c70002	L	-	-	0 (false)
CFG-RINV-DATA_SIZE	0x20c70003	U1	-	-	22
CFG-RINV-CHUNK0	0x50c70004	X8	-	-	0x203a656369746f4e ("Notice: ")
CFG-RINV-CHUNK1	0x50c70005	X8	-	-	0x2061746164206f6e ("no data ")
CFG-RINV-CHUNK2	0x50c70006	X8	-	-	0x000021646576173 ("saved!\0\0")
CFG-RINV-CHUNK3	0x50c70007	X8	-	-	0x0000000000000000

Satellite Systems (GNSS) Signal Configuration (CFG-SIGNAL-*) Configuration Defaults

Configuration Item	Key ID	Type	Scale	Unit	Default Value
CFG-SIGNAL-GPS_ENA	0x1031001f	L	-	-	1 (true)
CFG-SIGNAL-GPS_L1CA_ENA	0x10310001	L	-	-	1 (true)
CFG-SIGNAL-GPS_L2C_ENA	0x10310003	L	-	-	1 (true)
CFG-SIGNAL-GAL_ENA	0x10310021	L	-	-	1 (true)
CFG-SIGNAL-GAL_E1_ENA	0x10310007	L	-	-	1 (true)
CFG-SIGNAL-GAL_E5B_ENA	0x1031000a	L	-	-	1 (true)
CFG-SIGNAL-BDS_ENA	0x10310022	L	-	-	1 (true)
CFG-SIGNAL-BDS_B1_ENA	0x1031000d	L	-	-	1 (true)
CFG-SIGNAL-BDS_B2_ENA	0x1031000e	L	-	-	0 (false)
CFG-SIGNAL-QZSS_ENA	0x10310024	L	-	-	1 (true)
CFG-SIGNAL-QZSS_L1CA_ENA	0x10310012	L	-	-	1 (true)
CFG-SIGNAL-QZSS_L2C_ENA	0x10310015	L	-	-	1 (true)
CFG-SIGNAL-GLO_ENA	0x10310025	L	-	-	1 (true)
CFG-SIGNAL-GLO_L1_ENA	0x10310018	L	-	-	1 (true)
CFG-SIGNAL-GLO_L2_ENA	0x1031001a	L	-	-	1 (true)

Configuration of the SPI Interface (CFG-SPI-*) Configuration Defaults

Configuration Item	Key ID	Type	Scale	Unit	Default Value
CFG-SPI-MAXFF	0x20640001	U1	-	-	50
CFG-SPI-CPOLARITY	0x10640002	L	-	-	0 (false)
CFG-SPI-CPHASE	0x10640003	L	-	-	0 (false)
CFG-SPI-EXTENDEDTIMEOUT	0x10640005	L	-	-	0 (false)
CFG-SPI-ENABLED	0x10640006	L	-	-	0 (false)

Input Protocol Configuration of the SPI Interface (CFG-SPIINPROT-*) Configuration Defaults

Configuration Item	Key ID	Type	Scale	Unit	Default Value
CFG-SPIINPROT-UBX	0x10790001	L	-	-	1 (true)
CFG-SPIINPROT-NMEA	0x10790002	L	-	-	1 (true)
CFG-SPIINPROT-RTCM3X	0x10790004	L	-	-	1 (true)

Output Protocol Configuration of the SPI Interface (CFG-SPIOUTPROT-*) Configuration Defaults

Configuration Item	Key ID	Type	Scale	Unit	Default Value
CFG-SPIOUTPROT-UBX	0x107a0001	L	-	-	1 (true)
CFG-SPIOUTPROT-NMEA	0x107a0002	L	-	-	1 (true)
CFG-SPIOUTPROT-RTCM3X	0x107a0004	L	-	-	1 (true)

Time Mode Configuration (CFG-TMODE-*) Configuration Defaults

Configuration Item	Key ID	Type	Scale	Unit	Default Value
CFG-TMODE-MODE	0x20030001	E1	-	-	0 (DISABLED)
CFG-TMODE-POS_TYPE	0x20030002	E1	-	-	0 (ECEF)
CFG-TMODE-ECEF_X	0x40030003	I4	-	cm	0
CFG-TMODE-ECEF_Y	0x40030004	I4	-	cm	0
CFG-TMODE-ECEF_Z	0x40030005	I4	-	cm	0
CFG-TMODE-ECEF_X_HP	0x20030006	I1	0.1	mm	0
CFG-TMODE-ECEF_Y_HP	0x20030007	I1	0.1	mm	0
CFG-TMODE-ECEF_Z_HP	0x20030008	I1	0.1	mm	0

Time Mode Configuration (CFG-TMODE-) Configuration Defaults continued*

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Default Value*
CFG-TMODE-LAT	0x40030009	I4	1e-7	deg	0
CFG-TMODE-LON	0x4003000a	I4	1e-7	deg	0
CFG-TMODE-HEIGHT	0x4003000b	I4	-	cm	0
CFG-TMODE-LAT_HP	0x2003000c	I1	1e-9	deg	0
CFG-TMODE-LON_HP	0x2003000d	I1	1e-9	deg	0
CFG-TMODE-HEIGHT_HP	0x2003000e	I1	0.1	mm	0
CFG-TMODE-FIXED_POS_ACC	0x4003000f	U4	0.1	mm	0
CFG-TMODE-SVIN_MIN_DUR	0x40030010	U4	-	s	0
CFG-TMODE-SVIN_ACC_LIMIT	0x40030011	U4	0.1	mm	0

Tx-Ready Configuration (CFG-TXREADY-*) Configuration Defaults

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Default Value*
CFG-TXREADY-ENABLED	0x10a20001	L	-	-	0 (false)
CFG-TXREADY-POLARITY	0x10a20002	L	-	-	0 (false)
CFG-TXREADY-PIN	0x20a20003	U1	-	-	0
CFG-TXREADY-THRESHOLD	0x30a20004	U2	-	-	0
CFG-TXREADY-INTERFACE	0x20a20005	E1	-	-	0 (I2C)

Configuration of the UART1 Interface (CFG-UART1-*) Configuration Defaults

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Default Value*
CFG-UART1-BAUDRATE	0x40520001	U4	-	-	38400
CFG-UART1-STOPBITS	0x20520002	E1	-	-	1 (ONE)
CFG-UART1-DATABITS	0x20520003	E1	-	-	0 (EIGHT)
CFG-UART1-PARITY	0x20520004	E1	-	-	0 (NONE)
CFG-UART1-ENABLED	0x10520005	L	-	-	1 (true)
CFG-UART1-REMAP	0x10520006	L	-	-	0 (false)

Input Protocol Configuration of the UART1 Interface (CFG-UART1INPROT-*) Configuration Defaults

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Default Value*
CFG-UART1INPROT-UBX	0x10730001	L	-	-	1 (true)
CFG-UART1INPROT-NMEA	0x10730002	L	-	-	1 (true)
CFG-UART1INPROT-RTCM3X	0x10730004	L	-	-	1 (true)

Output Protocol Configuration of the UART1 Interface (CFG-UART1OUTPROT-*) Configuration Defaults

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Default Value*
CFG-UART1OUTPROT-UBX	0x10740001	L	-	-	1 (true)
CFG-UART1OUTPROT-NMEA	0x10740002	L	-	-	1 (true)
CFG-UART1OUTPROT-RTCM3X	0x10740004	L	-	-	1 (true)

Configuration of the UART2 Interface (CFG-UART2-*) Configuration Defaults

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Default Value*
CFG-UART2-BAUDRATE	0x40530001	U4	-	-	38400
CFG-UART2-STOPBITS	0x20530002	E1	-	-	1 (ONE)
CFG-UART2-DATABITS	0x20530003	E1	-	-	0 (EIGHT)
CFG-UART2-PARITY	0x20530004	E1	-	-	0 (NONE)
CFG-UART2-ENABLED	0x10530005	L	-	-	1 (true)

Configuration of the UART2 Interface (CFG-UART2-) Configuration Defaults continued*

Configuration Item	Key ID	Type	Scale	Unit	Default Value
CFG-UART2-REMAP	0x10530006	L	-	-	0 (false)

Input Protocol Configuration of the UART2 Interface (CFG-UART2INPROT-*) Configuration Defaults

Configuration Item	Key ID	Type	Scale	Unit	Default Value
CFG-UART2INPROT-UBX	0x10750001	L	-	-	1 (true)
CFG-UART2INPROT-NMEA	0x10750002	L	-	-	1 (true)
CFG-UART2INPROT-RTCM3X	0x10750004	L	-	-	1 (true)

Output Protocol Configuration of the UART2 Interface (CFG-UART2OUTPROT-*) Configuration Defaults

Configuration Item	Key ID	Type	Scale	Unit	Default Value
CFG-UART2OUTPROT-UBX	0x10760001	L	-	-	1 (true)
CFG-UART2OUTPROT-NMEA	0x10760002	L	-	-	1 (true)
CFG-UART2OUTPROT-RTCM3X	0x10760004	L	-	-	1 (true)

Configuration of the USB Interface (CFG-USB-*) Configuration Defaults

Configuration Item	Key ID	Type	Scale	Unit	Default Value
CFG-USB-ENABLED	0x10650001	L	-	-	1 (true)
CFG-USB-SELFPOW	0x10650002	L	-	-	1 (true)
CFG-USB-VENDOR_ID	0x3065000a	U2	-	-	5446
CFG-USB-PRODUCT_ID	0x3065000b	U2	-	-	425
CFG-USB-POWER	0x3065000c	U2	-	mA	0
CFG-USB-VENDOR_STR0	0x5065000d	X8	-	-	0x4120786f6c622d75 ("u-blox A")
CFG-USB-VENDOR_STR1	0x5065000e	X8	-	-	0x2e777777202d2047 ("G - www.")
CFG-USB-VENDOR_STR2	0x5065000f	X8	-	-	0x632e786f6c622d75 ("u-blox.c")
CFG-USB-VENDOR_STR3	0x50650010	X8	-	-	0x0000000000006d6f ("om\0\0\0\0\0\0")
CFG-USB-PRODUCT_STR0	0x50650011	X8	-	-	0x4720786f6c622d75 ("u-blox G")
CFG-USB-PRODUCT_STR1	0x50650012	X8	-	-	0x656365722053534e ("NSS rece")
CFG-USB-PRODUCT_STR2	0x50650013	X8	-	-	0x0000000072657669 ("iver\0\0\0\0")
CFG-USB-PRODUCT_STR3	0x50650014	X8	-	-	0x0000000000000000
CFG-USB-SERIAL_NO_STR0	0x50650015	X8	-	-	0x0000000000000000
CFG-USB-SERIAL_NO_STR1	0x50650016	X8	-	-	0x0000000000000000
CFG-USB-SERIAL_NO_STR2	0x50650017	X8	-	-	0x0000000000000000
CFG-USB-SERIAL_NO_STR3	0x50650018	X8	-	-	0x0000000000000000

Input Protocol Configuration of the USB Interface (CFG-USBINPROT-*) Configuration Defaults

Configuration Item	Key ID	Type	Scale	Unit	Default Value
CFG-USBINPROT-UBX	0x10770001	L	-	-	1 (true)
CFG-USBINPROT-NMEA	0x10770002	L	-	-	1 (true)
CFG-USBINPROT-RTCM3X	0x10770004	L	-	-	1 (true)

Output Protocol Configuration of the USB Interface (CFG-USBOUTPROT-*) Configuration Defaults

Configuration Item	*Key ID*	*Type*	*Scale*	*Unit*	*Default Value*
CFG-USBOUTPROT-UBX	0x10780001	L	-	-	1 (true)
CFG-USBOUTPROT-NMEA	0x10780002	L	-	-	1 (true)
CFG-USBOUTPROT-RTCM3X	0x10780004	L	-	-	1 (true)

Related Documents

Overview

As part of our commitment to customer support, u-blox maintains an extensive volume of technical documentation for our products. In addition to product-specific data sheets and integration manuals, general documents are also available. These include:

- GPS Compendium, Docu. No GPS-X-02007
- GPS Antennas - RF Design Considerations for u-blox GPS Receivers, Docu. No GPS-X-08014

Our website www.u-blox.com is a valuable resource for general and product specific documentation.

For design and integration projects the Receiver Description Including Interface Description should be used together with the Data Sheet and Hardware Integration Manual of the GNSS receiver.

Related Documents for Chips and Chipsets

u-blox chipset documentation requires an NDA. Contact u-blox for the following documents:

Related Documents for Modules

Documentation for some products can be downloaded from our website. For other products please contact u-blox.

Revision History

Revision	*Date*	*Name*	*Status / Comments*
R01	23-May-2018	pkeh / jhak	initial version
R02	25-May-2018	jhak	Added message UBX-MON-TEMP
R03	21-Jun-2018	jhak	sigID field for UBX-RXM-RAWX
R04	06-Jul-2018	jhak	ES TBD

Contact

For complete contact information visit us at www.u-blox.com

u-blox Offices

North, Central and South America

u-blox America, Inc.
Phone: +1 703 483 3180
E-mail: info_us@u-blox.com

Regional Office West Coast:
Phone: +1 408 573 3640
E-mail: info_us@u-blox.com

Technical Support:
Phone: +1 703 483 3185
E-mail: support_us@u-blox.com

Headquarters
Europe, Middle East, Africa

u-blox AG
Phone: +41 44 722 74 44
E-mail: info@u-blox.com
Support: support@u-blox.com

Documentation Feedback
E-mail: docsupport@u-blox.com

Asia, Australia, Pacific

u-blox Singapore Pte. Ltd.
Phone: +65 6734 3811
E-mail: info_ap@u-blox.com
Support: support_ap@u-blox.com

Regional Office Australia:
Phone: +61 2 8448 2016
E-mail: info_anz@u-blox.com
Support: support_ap@u-blox.com

Regional Office China (Beijing):
Phone: +86 10 68 133 545
E-mail: info_cn@u-blox.com
Support: support_cn@u-blox.com

Regional Office China (Chongqing):
Phone: +86 23 6815 1588
E-mail: info_cn@u-blox.com
Support: support_cn@u-blox.com

Regional Office China (Shanghai):
Phone: +86 21 6090 4832
E-mail: info_cn@u-blox.com
Support: support_cn@u-blox.com

Regional Office China (Shenzhen):
Phone: +86 755 8627 1083
E-mail: info_cn@u-blox.com
Support: support_cn@u-blox.com

Regional Office India:
Phone: +91 80 4050 9200
E-mail: info_in@u-blox.com
Support: support_in@u-blox.com

Regional Office Japan (Osaka):
Phone: +81 6 6941 3660
E-mail: info_jp@u-blox.com
Support: support_jp@u-blox.com

Regional Office Japan (Tokyo):
Phone: +81 3 5775 3850
E-mail: info_jp@u-blox.com
Support: support_jp@u-blox.com

Regional Office Korea:
Phone: +82 2 542 0861
E-mail: info_kr@u-blox.com
Support: support_kr@u-blox.com

Regional Office Taiwan:
Phone: +886 2 2657 1090
E-mail: info_tw@u-blox.com
Support: support_tw@u-blox.com